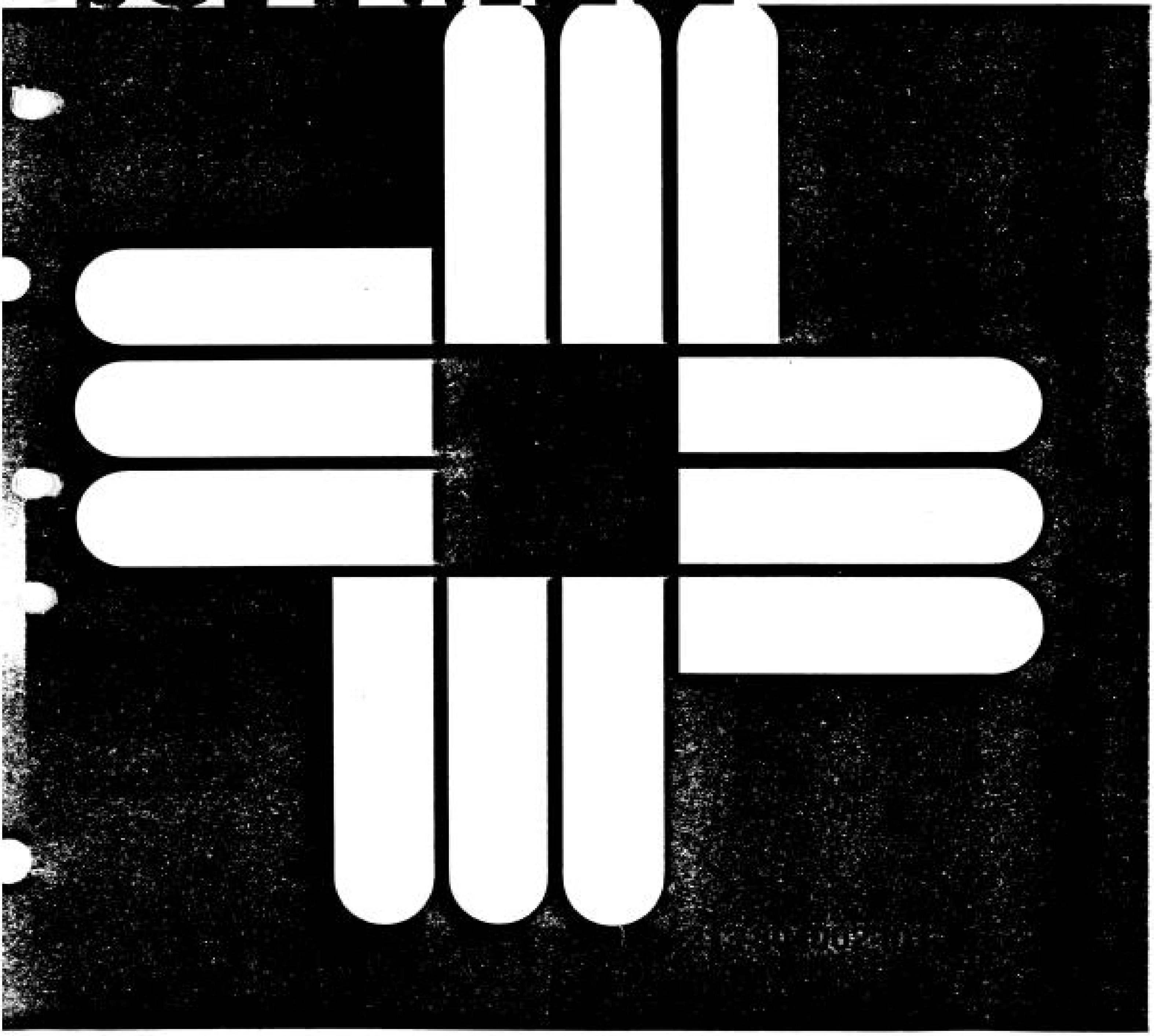
MODCOMP
MODULAR COMPUTER SYSTEMS, INC.

MAX IV System Guide Manual

SYSGEN

SOFTWARE



MODULAR COMPUTER SYSTEMS, INC.

System Guide Manual
MAX IV

SYSGEN

MANUAL HISTORY

Manual Order Number: 213-804002-I01

Title: MAX IV SYSTEM GUIDE MANUAL, SYSGEN

Software Part Number: 610304-000I.1

Model Number: 8240I

Revision Level	**Date Issued**	**Description**
H00	03/83	Initial Issue (H.0).
H01	11/83	Revision (H.1). Included information for MAX IV OS Revision H.1.
H02	06/84	Reissue (H.2). Include information for MAX IV OS Revision H.2.
I00	09/84	Reissue (I.0). Changes include a TASKMASTER Directive enhancement, the addition of the TOCLOCK Directive, and the removal of the WORKBENCH Directive.
I01	09/85	Revision (I.1). Changes include new default values for the TASK statement, a changed device default for the standard keyboard/screen device, and additional clarity for the TOCLOCK and CONTROLLER statements. DCS information was included.

Contents subject to change without notice.

PROPRIETARY INFORMATION

THIS MANUAL CONTAINS CONFIDENTIAL PROPRIETARY INFORMATION PROTECTED BY SOFTWARE PROPERTY RIGHTS AND IS MADE AVAILABLE UPON THE CONDITION THAT THE SOFTWARE CONTAINED HEREIN WILL BE HELD IN ABSOLUTE CONFIDENCE AND MAY NOT BE DISCLOSED IN WHOLE OR IN PART TO OTHERS WITHOUT THE PRIOR WRITTEN PERMISSION OF MODULAR COMPUTER SYSTEMS, INC.

FOR GOVERNMENT USE THE FOLLOWING SHALL APPLY:

RESTRICTED RIGHTS LEGEND

USE, DUPLICATION OR DISCLOSURE BY THE GOVERNMENT IS SUBJECT TO RESTRICTIONS AS SET FORTH IN PARAGRAPH (b) (3) (B) OF THE RIGHTS IN TECHNICAL DATA AND COMPUTER SOFTWARE CLAUSE IN DAR7-104.9(a) (OR SUCH EQUIVALENT FPR, NASPR, AND/OR DOEPR CLAUSE AS MAY BE APPLICABLE).

MODULAR COMPUTER SYSTEMS, INC.
1650 WEST McNAB ROAD
FORT LAUDERDALE, FL 33309

PREFACE

Audience

The information in this manual is directed to the system manager or system programmer responsible for configuring the MAX IV Operating System.

Subject

This manual contains information about each statement in the system generation (SYSGEN) process. A detailed description of each SYSGEN statement's format and parameters, both required and optional is included. In addition, examples illustrating the usage of individual statements are given.

Those users familiar with previous issues of MAX IV Operating Systems manuals may notice that the SYSGEN information was previously contained in Chapter 8 of the MAX IV GENERAL OPERATING SYSTEM, Reference Manual (210-610304-000).

Products Supported

MAX IV Software System, Model 8240I.

Special Symbols and Notations

A revision bar (|) located in the margin of the page in the text indicates a change to the manual. This change generally represents a technical change to the product due to product revision. A revision bar is also entered in the Table of Contents to flag the general location of changes in the text.

Related Publications

The reader is referred to the manuals listed below for additional information. When ordering manuals, use the Manual Order Number listed below. The latest revision (REV) will be shipped.

Manual Order Number	Manual Title
213-804001-REV	MAX IV GENERAL OPERATING SYSTEM System Guide Manual (SGM)
213-804003-REV	MAX IV EXECUTIVE (REX) SERVICES System Guide Manual (SGM)
213-804004-REV	MAX IV OPERATOR COMMUNICATIONS System Guide Manual (SGM)
213-804005-REV	MAX IV BASIC INPUT/OUTPUT SYSTEM System Guide Manual (SGM)
213-804006-REV	MAX IV UNIT RECORD DEVICE HANDLERS System Guide Manual (SGM)

211-804009-REV	MAX IV LINK EDITOR (EDI) Programmer's Reference Manual (PRM)
213-804007-REV	MAX IV DATA STORAGE DEVICE HANDLERS System Guide Manual (SGM)
213-804008-REV	MAX IV COMMUNICATION HANDLERS System Guide Manual (SGM)
211-804005-REV	MAX IV/MAX 32 SOURCE EDITOR Programmer's Reference Manual (PRM)
211-804001-REV	MAX IV and MAX 32 TASK/OVERLAY CATALOGER Programmer's Reference Manual (PRM)
210-804001-REV	MAX IV/MAX 32 MODCOMP ASSEMBLERS Language Reference Manual (LRM)

MAX IV REVISION I.1 SUMMARY

For MAX IV Rev. I.1, the TASK statement default values were changed to 255, and the standard non-disc device default for the keyboard/printer or keyboard/screen was changed to 82 bytes. Documentation changes were made to the TOCLOCK statement and the CONTROLLER statement for clarity. New information was provided for the Distributed Communications Subsystem (DCS).

MAX IV REVISION I.Ø SUMMARY

The only necessary change to a MAX IV SYSGEN source file concerns the WORKBENCH feature. If WORKBENCH is not running, no SYSGEN changes are required.

For MAX IV Rev. I.Ø, the WORKBENCH statement must be removed from Part Two of the SYSGEN source, Basic I/O Section. The WORKBENCH statement in Part One of the SYSGEN is all that is required, for Rev. I.Ø. Leaving the second WORKBENCH statement in the source causes an assembler error (Opcode error).

SYSGEN changes are necessary to take advantage of many of the features discussed below.

TOCLOCK - ENABLE/DISABLE TOCLOCK

The TOCLOCK feature, when enabled, attempts to protect the integrity of load module files during all activities that affect the load module directory. Concurrent access of a load module file is guarded, and the module loader is prevented from loading a module that is currently being manipulated by TOC. TOCLOCK attempts to provide the user with the ability to safely perform on-line compression and load module file updates, without jeopardizing system integrity. This feature and its related data structures can be eliminated if it is not desired or needed.

TASKMASTER - ROLL-ON-HOLD OPTION

In a multibatch environment, the user may want to roll out tasks that are in I/O HOLD, OC HOLD or are suspended. Parameter 3 of the TASKMASTER statement, when used in conjunction with the ROUNDROBIN option (Parameter 2) provides this capability. This feature affects only those tasks that are cataloged by the Task/Overlay Catalogar as WORKBENCH or PECULIAR BATCH.

AUTO-REBOOT STAND-ALONE LINKING LOADER (ASAL)

With MAX IV Revision I.Ø, source of power fail/restart routines, including the Auto-Reboot Stand-Alone Linking Loader (ASAL) are included on USL.

Appendix H provides information to aid the user in implementing a power fail/restart sequence.

PROGRAMMABLE LOGIC CONTROLLER

Identifies the SYSGEN LDT option for a Programmable Logic Controller.

OCTAL ASYNCHRONOUS INTERFACE (OAI)

Identifies the controller class for the Octal Asynchronous Interface (OAI) model 48Ø9 and MODACS II model 18Ø6.

NOTE: The following revision summaries are provided for users updating from versions of MAX IV prior to revision H.Ø.

MAX IV REVISION H.Ø SUMMARY

The following SYSGEN statements have been added in Revision H.Ø of the MAX IV Operating System:

- MLSEQUENTIAL
- OCPASSWORD
- BRKCHARACTER
- DITPOOL
- LISTCONTROL
- LINEMONITOR
- LOGTRANSPORT

Sequential Task Loader

The MAX IV Sequential Task Loader provides the user with all the task activation capabilities of the current module loader at system start-up. It permits the user to activate tasks stored on sequential media rather than on a load module file.

To include the MAX IV Sequential Task Loader rather than the MAX IV Module Loader in a system, the MLSEQU statement must be included in the system block of the SYSGEN.

The PRESCHEDULE statement must not be used in systems using the Sequential Task Loader.

Operator Communications Password

A global password feature is available for all remote terminal users using TMP. Access to the Operator Communications Task can be restricted by password or made unavailable to remote terminal users with the OCPASSWORD SYSGEN statement.

System Break Character

The BRKCHARACTER statement can be used in a SYSGEN to define the system break character. If the statement is not used, the break character defaults to CONTROL "A".

System Dynamic Interval Timers

The DITPOOL statement can be used in a SYSGEN to specify the number of system dynamic interval timers.

Terminal Listing Control

Terminal Listing Control (TLC) allows the terminal user to hold or resume output to the terminal screen. This capability is selected at SYSGEN in the option parameter of the COMDEVICE statement.

When TLC is enabled, the system recognizes default listing control characters. The LISTCONTROL SYSGEN statement may be used to define special listing control characters for the listing stop or listing resume functions. An automatic listing resume time interval may also be specified on the LISTCONTROL statement.

Ring Detect Support

Ring Detect Support provides the user with an auto answer mechanism that may be tailored to suit most system environments. Auto task activation (connect) and task deestablish/kill (disconnect) capabilities may be optionally selected. Hooks are available within the Terminal Monitor Program (TMP) that enable the user to perform special functions on detection of RING, BREAK, and HANG UP events.

Ring Detect Support gives the user a mechanism that allows remote connection to a MODCOMP communications line. Support includes asynchronous, byte synchronous and bit synchronous line interfaces.

The new LINEMONITOR SYSGEN statement is used to configure communication channels as either Break character/Key supported and/or Ring supported.

Logical Transport

The LOGTRANSPORT statement is used to identify the logical transport(s) for each physical transport. There must be at least one LOGTRAN statement for every FMTRANSPORT statement.

The following SYSGEN statements have been modified in Revision H.Ø of the MAX IV Operating System:

- INTERTASK
- CALARM
- MACHINE
- SYSBUG
- FMTRANSPORT
- IONODES
- ROLLER

INTERTASK

Parameter 3 specifies that ITC will be run in a map other than MAP Ø.

CALARM

The influence limit is now specified as Parameter 4. Parameter 3 specifies the setting of trigger bits.

MACHINE

The MODCOMP IV hardware is no longer supported. All code related to the MODCOMP IV has been removed from MAX IV.

Actual Memory Dump

The Actual Memory Dump Package can be included in the system with the SYSBUG statement in the SYSGEN.

Physical Transport

The FMTRANSPORT statement is used to identify the physical transports for each controller. This statement is used in place of the DISCTRANSPORT statement. The DISCTRANSPORT statement is no longer supported in MAX IV as of Revision H.Ø.

IONODES

Parameter 3 allows the user to specify the maximum number of global nodes that can ever be allocated.

ROLLER

The user can configure the NON-MAP Ø resident roller in the system by specifying the keyword MAPPED as Parameter 2 of the ROLLER SYSGEN statement. The file from which the NON-MAP Ø resident roller is to be loaded can be specified as Parameter 3.

The following SYSGEN statements cannot be used with MAX IV Revision H.Ø:

BREAK

> The old BREAK statement generates a SYSGEN error under MAX IV, Revision H.Ø. The BREAK statement's optional password argument can be specified with the new OCPASSWORD SYSGEN statement. Refer to the LINEMONITOR statement.

MINPRIVATE

DISCTRANSPORT

> The old DISCTRANSPORT statement generates a SYSGEN error under MAX IV Revision H.Ø. Systems should substitute an appropriate FMTRANSPORT statement for each occurrence of the DISCTRANSPORT statement.

MAX IV REVISION H.1 SUMMARY

The Revision H.1 release of this manual includes information concerning the 4185 disc subsystem. Additional information is provided for the usage of MAXPAGES on the SHPRIV statement.

Additional information on GLOBAL statement usage is given, as well as SYSGEN updates in Appendix J.

MAX IV REVISION H.2 SUMMARY

The Revision H.2 release of this manual includes additional information concerning the MLSEQ and OCPASS statements. SYSGEN updates were added in Appendix F, and further updates were added to Appendix I sections discussing Resident OC Modules and Resident OC Packages.

TABLE OF CONTENTS

CHAPTER 1
OVERVIEW

A MAX IV Operating System can be configured so that only the features required for a particular application are included. Preparation of a custom MAX IV is similar to the preparation of any program written in a high-level language. That is:

- Statements are written that define all data structures and procedures.

- These statements are processed by a language-translation program and are transformed into an incomplete object module (main program).

- A link editor finds missing subroutines by searching a library of optional subroutines. It links the main program to the subroutines called and forms a complete program load module.

- Data for the program to process is prepared on external media (for example, disc).

- The load module is loaded and executed and processes its data.

In the case of MAX IV System Generation (SYSGEN), the complete-load-module is the resident nucleus of the operating system. The data-to-be-processed includes the non-resident programs and tasks to be executed by the operating system. The library-of-optional-subroutines includes the various modular elements of the operating system that are already assembled in object format. This library is called the System Elements Library. The data structures of the operating system and all calls to optional elements are defined by a macro language that is processed by the Macro Assembler.

1.1 SYSTEM GENERATION DESCRIPTION

The System Generation process is described in the following sections in terms of the three basic phases:

1. Macro Language Phase

2. Linking Phase

3. Loading and Initialization Phase

1.1.1 PHASE 1 MACRO LANGUAGE

This phase requires that the user specify, in an easily understood macro language, which of the optional elements of the system are required for the particular application. The macro language statements are processed by the Macro Assembler. One or more incomplete binary object modules result. They contain:

- The fixed data base (lists, tables, stacks, and so on) that defines all operating system structures.
- External calls to all selected subroutines (handlers, services, and so on) in the System Elements Library.
- Calls to all custom user-coded elements to be made part of the resident system nucleus (special interrupt routines, tables, and so on).

These object programs are collectively called the SYSBLOCK and they act as the main program of the operating system. The SYSBLOCK is produced as several separate object modules (instead of one large object module) in the interest of requiring less memory to process it. The parts are given the following names:

PART 1 - THE SYSTEM STRUCTURE BLOCK
PART 2 - THE I/O STRUCTURE BLOCK
PART 3 - THE TASK STRUCTURE BLOCK
PART 4 - THE FILE MANAGER STRUCTURE BLOCK
PART 5 - USER-CODED RESIDENT ELEMENTS

In the next phase, these parts are linked to other subroutines contained in object libraries.

1.1.2 PHASE 2 LINKING

This phase links all the object elements of the various parts of the operating system nucleus. It is performed by either:

- Stand-Alone Linking Loader. If the user can allow the time required for the linking process each time the system is loaded, then Phases 2 and 3 can be combined by using the Stand-Alone Linking Loader (SAL). This program links as it loads.
- Link Editor (EDI). The user can use the Link Editor (EDI) if a copy of the complete system is desired. This edited object system is loaded by the Stand-Alone Loader (SAL) and does not require linking for each load. A hard copy link edit map may optionally be produced. All modules listed in the link edit map up to RS.VIN will remain in virtual MAP Ø after COLD START.

NOTE: If any routines appear in the SYSGEN map after CS.INI, the system is unusable. Reorganize the System Elements Library and then reassemble.

1.1.3 PHASE 3 LOADING AND INITIALIZATION

This phase loads a complete system load module into memory and transfers control to this new operating system to start normal system operation.

Phase 3 includes the execution of the start-up routines of the operating system. The COLD START routines are no longer required once the system has been started, and the memory used for these routines is returned to memory pools. The RESTART routine survives in the system and is executed whenever a power failure or Master-Clear occurs.

If Phase 2 is performed using the Stand-Alone Linking Loader, then most of Phase 3 is completed early. Phase 3 is performed by the Stand-Alone Linking Loader in either case - even if it simply loads an already prepared (and fully linked) load module of the system.

The COLD START and RESTART routines are performed in the Non-Virtual Mode, before the CLASSIC enters the Virtual Mode.

During start-up a halt may occur. The halt address may be checked by referring to the SYSGEN link edit map. The system may halt at internal MM$NOP during X task allocation of MAP O pages for I/O nodes or other requirements. This halt occurs when no additional MAP Ø pages are available for allocation. The system must be regenerated to reduce its size. The user should eliminate unused data structures, reduce the number of I/O nodes and delete other unnecessary modules.

If the X task attempts to abort during start-up and sense switch Ø on the control panel is reset and VDUMP is configured in the system, a dump of MAP Ø will be generated.

CHAPTER 2
SYSGEN STATEMENTS

The following sections explain each System Generation (SYSGEN) statement including:

- The activity it causes
- The rules to follow when using it

These statements are all involved in Phase 1 (the Macro Language Phase) as described in Chapter 1.

MAX IV SYSGEN specification statements are grouped into the following parts:

PART 1 - SYSTEM STRUCTURE BLOCK

PART 2 - BASIC I/O STRUCTURE BLOCK

PART 3 - TASK STRUCTURE BLOCK

PART 4 - FILE MANAGER STRUCTURE BLOCK

PART 5 - USER-CODED RESIDENT ELEMENTS

The source file for the SYSGEN assembly phase must contain the selected statements in the order in which they are presented in this chapter. Some statements are required, some statements are optional. Some statements may be specified only once, still others may be repeated as necessary. All statements have a keyword and a list of optional or required parameters.

The keyword normally starts in column 7 although any position between column 2 and column 21 is acceptable. There must be at least one space between the end of the keyword and the start of the first parameter. The first parameter typically starts in column 21. Parameters are separated by a comma.

In the parameter string, if a parameter is allowed to default and there are additional parameters to specify, the default parameter must be indicated in the parameter list. This default parameter is indicated with the parameter terminating character, a comma.

The format used in the presentation of this information is as follows:

```
KEYWORD

-----------------------------------------------------------------
BRIEF SUMMARY
-----------------------------------------------------------------

Description in paragraph form.

SYNTAX

                              1         2
     KEYWORD             parameter1 parameter2

parameter1          - (Description in paragraph form)

parameter2          - (Description in paragraph form)

EXAMPLES
```

Symbols used:

xxx — Lower-case letters represent a variable, the value of which is supplied by the user.

XXX — Upper-case keywords are to be entered as shown.

[ ] — Brackets indicate either insignificant characters or an optional parameter.

{ } — Braces indicate a set of options from which the user is required to choose at least one.

LISTING TITLE - START OF NEW ASSEMBLY

The TTL statement is a standard Macro Assembler statement. Use of a TTL statement is optional. It can contain any message. It ensures that the SYSGEN listing starts on a whole page boundary -- and that the title specified appears on each page of the listing.

SYNTAX

```
                        1
     TTL             heading
```

heading - Parameter 1 indicates the heading that appears at the top of each page of this part of the SYSGEN listing.

EXAMPLE

```
     TTL   MAX IV: SYSBLK
```

INSERT MACRO PROTOTYPES - TO DEFINE MACRO-LANGUAGE

Before the macro language that defines MAX IV system structures can be used, the prototypes that define the language must be inserted from an external file. This standard Macro Assembler statement performs this function and MUST be present.

SYNTAX

```
                      1     2
     INS             fname IVSYS
```

fname - Parameter 1 specifies the logical file name where the macro prototypes are cataloged. The MAX IV standard for this file is MC, the Common Macro Library. This file name must be assigned to a real file or device before it can be used.

IVSYS - Parameter 2 is the name of the macro-definition file used to define the statements in the System Structure Block.

EXAMPLE (only one use permitted per assembly)

```
     INS   MC,IVSYS
```

All the SYSGEN statements in PART 1 are defined by the MAX IV System Structure Prototypes. These definitions are included in the assembly by using this statement.

NAME OF PROGRAM

The PGM statement is a standard Macro Assembler statement. Use of a PGM statement is optional. It assigns a name to the program being assembled. This statement must follow the INS statement in order to override the standard default provided by the macro file inserted for this block.

SYNTAX

```
                          1
     PGM                name
```

name - Parameter 1 is a valid symbolic name as defined in the MODCOMP Assemblers, Language Reference Manual. GS.IV is the default name for the System Structure Block.

SPECIFY TASKMASTER OPTIONS

The TASKMASTER statement is optional, however, the statement is implicitly generated by the ENDSYSTEM statement if the user does not specify it. When generated in this manner, all parameters are defaulted.

MAX IV Taskmaster context switches are event driven. The TASKMASTER statement defines a period of time as an event. If no natural context switch occurs during the specified time interval one will be forced on expiration of a directly connected timer. The timer is reset by the Taskmaster on event scans.

The TASKMASTER statement influences the standard system context switching routine (TASKMASTER). Normally, the Taskmaster processes tasks strictly on the basis of priority. Even if several tasks exist in the CPU Queue (CPUQ) in clusters at the same priority level, the tasks closer to the head of the CPUQ will actually be higher in priority than the tasks of the same priority level that are lower in the CPUQ.

Specifying a ROUNDROBIN scheduling algorithm for tasks of like priority level results in a more even distribution of computer time to tasks of equal priority when they each have a tendency to become "compute bound". This algorithm moves the task using the most CPU time lower in the CPU queue. Task movement in the CPU queue is with respect to priority clusters. High priority tasks in the CPU queue are always given preference over lower priority tasks regardless of CPU utilization.

SYNTAX

```
                          1          2                 3              4
    TASKMA[STER]       [rate] [ROU[NDROBIN]] [ROL[ONHOLD]] [interval]
```

[rate] - Parameter 1 is optional. It specifies a period of time that is allowed to pass before an event scan is forced by the real-time clock. The time period is defined by a number of basic clock interrupts (at 200 Hz). The timer is reset by the Taskmaster or event scans.

If the value specified is zero (0), no forced context switching occurs and only natural events drive the system's context switching function. The default value is 100ms (approximately 1/2 second).

[ROU[NDROBIN]] - Parameter 2 is optional. It specifies that the round-robin scheme of scheduling is used for groups of tasks in the CPUQ at the same priority level.

[ROL[ONHOLD]] - Parameter 3 is optional. When specified along with Parameter 2 (ROU) this parameter rolls out tasks that are suspended or are in I/O HOLD or OC HOLD. Parameter 3 only affects tasks that are cataloged through the Task/Overlay Cataloger (TOC) WORKBENCH or PECULIAR batch.

[interval] - Parameter 4 is optional and has meaning only if Parameter 2 (ROU) is also specified. This parameter specifies the number of basic clock interrupts (at 2ØØ HZ) in the interval to be used by the ROUNDROBIN algorithm to rearrange the CPUQ. If ROU is specified, but interval is not, then a default value of 1ØØ clock intervals (approximately 1/2 second) is used.

EXAMPLES

```
    TASKMASTER   1ØØ
         Standard rate.

    TASKMA   2ØØ,ROU
         1 second rate, ROUNDROBIN.

    TASKMASTER   1ØØ,ROU,,4ØØ
         Standard rate, ROUNDROBIN, rearrange every 2 seconds.

    TASKMA   ~~1~~2ØØ,ROU,ROL
         1  second  rate, ROUNDROBIN, roll out I/O or OC HOLDs or
         suspended tasks.
```

IIISER

INCLUDE MAX III COMPATIBLE EXECUTIVE (REX) SERVICES

The IIISERVICES statement is optional unless Job Control and any of the system processors are to be used. It specifies that compatible MAX III services are to be processed by the system being generated. MAX IV has many REX services that are called MAX-IV-type services because they have no inline memory arguments -- passing arguments exclusively in registers. MAX-III-type services, on the other hand, generally use inline memory arguments. Refer to the MAX IV EXECUTIVE (REX) SERVICES System Guide Manual (listed in the Preface) for more information about these services.

SYNTAX

```
     IIISER[VICES]
```

EXAMPLES

```
     IIISERVICES
          Specify MAX III services.

     IIISER
          Specify MAX III services.
```

SPECIFY INDIVIDUAL EXECUTIVE (REX) SERVICES

The SERVICE statement (optional if SERPAC statement is used) can be repeated as necessary to specify each optional REX service to be included in MAX IV. An optional REX service can be specified as part of a standard package name (see SERPACK statement) or by individual service name. The specified service's entry point appears in the MAX IV REX service table and is called "M$name". If the IIISERVICE statement has been specified, the service also has a second entry point that appears in the MAX III compatible table with the name "M$$nam" if the service number is less than #44. (The(se) name(s) are externally defined.)

SYNTAX

```
                        1      2              3
     SERVIC[E]          name [number]  [N[ONRESIDENT]]
```

name - Parameter 1 is the name (1- to 4-characters) of a standard MAX IV REX service or custom user-coded REX service. The names of standard system services can be selected from the list in Appendix B. Any user-coded service must have a name unique to these standard services.

[number] - Parameter 2 is optional. It specifies the number to be associated with a custom, user-coded REX service. Standard system REX service numbers need not be specified since their numbers are fixed. The value of this parameter must not duplicate a standard system number. The default is a function of the name given as Parameter 1.

[N[ONRESIDENT]] - Parameter 3 specifies the service to be non-resident. Only custom services are usable in the non-resident form. Refer to MAX IV Executive (REX) Services, System Guide Manual listed in the Preface.

EXAMPLES

```
     SERVICE   DUMP
          Specify standard hex dump service.

     SERVICE   MYSERV,#44
          Specify custom service.
```

SERPAC

SPECIFY PACKAGES OF EXECUTIVE SERVICES

The SERPACK statement (optional) can be used instead of the SERVICE statement to specify functional groups of REX services by group name instead of by specific service name. The use of this statement results in the internal calling of the SERVICE statement for each service belonging to the named package. Some packages call other packages and one particular package (ALL) calls for all services and packages.

SYNTAX

```
                          1              2
     SERPAC[K]       packagename [R[ESIDENT    ]]
                                 [N[ONRESIDENT]]
```

packagename - Parameter 1 specifies the standard package name that can be selected from the list in Appendix B.

[R[ESIDENT]]
[N[ONRESIDENT]] - Parameter 2 is optional and defaults to RESIDENT. It specifies whether the package called for in Parameter 1 is to be resident or non-resident. Resident services execute quickly but consume addressable space in MAP Ø (perhaps to the exclusion of some other service) while non-resident services execute slowly (because they must first be loaded from mass storage) but consume no addressable space while non-resident and share addressable space with other non-resident elements even when resident. Notice that when non-resident elements are specified, each use of such an element results in a copy of that element being loaded. The system maintains no record of which non-resident elements are currently in memory and, in fact, if two (or more) tasks make calls to the same non-resident service, each call executes a separate copy of the service even though the service is re-entrant.

None of the standard MAX IV services may be non-resident; however, the user may add any number of user-coded non-resident services.

EXAMPLES

```
     SERPACK   ALL
          Specify all services.

     SERPAC   BIOS
          Specify basic I/O system.

     SERPAC   XIOS
          Specify extended I/O system.

     SERPACK   BTASK
          Specify basic task services.
```

INTERTASK

DEFINE ITC IN THE SYSTEM

The INTERTASK statement is optional. It defines the ITC (Intertask Communication System) REX services in the system being generated. ITC system defaults are also defined using this statement.

The Intertask REX Services are not included in the system as a result of a call to SERPAC ALL. Explicit reference to this statement is required to include ITC.

NOTE: This statement generates an implicit reference to the SERVICE macro #42 on the ITC REX call number.

SYNTAX

```
                           1        2          3
     INTERT[ASK]       [pages] [sysdws] [MAP[PED]]
```

[pages] - Parameter 1 is optional. It represents the maximum number of actual memory pages that may be used at any given time by ITC. The total number of actual pages used by ITC includes:

- o Actual memory used for messages
- o Port
- o Server definitions
- o Port descriptors not defined in the SYSGEN.

The default is 200.

[sysdws] - Parameter 2 is optional. It represents the default window size to be used by tasks not implicitly specifying a window size when using ITC. The default is 8 message nodes. A message node is 248 words.

[MAP[PED]] - Parameter 3 is specified only if ITC is to be run in a map other than MAP 0. Refer to the MAX IV GENERAL OPERATING SYSTEM, System Guide Manual listed in the Preface, for a complete explanation of how to build a non-MAP 0 resident ITC.

EXAMPLES

```
     INTERTASK
          Enter ITC into the system, permit 200 actual pages to be
          used at any time, and set the default window size to 8.

     INTERTASK 500,20
          Enter ITC into the system, permit 500 actual pages to be
          used at any time, and set the default window size to 20.
```

DEFINE AN ITC PORT

The PORT statement is optional, and if entered, defines an ITC (Intertask Communication System) port structure for the task specified. A task may have only one port structure, although more than one SERVER statement may refer to the task's port. Therefore, for a given task, it is only necessary to enter one PORT statement. The port created has meaning only if the INTERTASK macro is also used.

SYNTAX

```
                              1         2     3       4          5
     PORT              task-name [pri] [lmf] [maxvcs] [options]

                           6         7
                       [options] [options]
```

task-name - Parameter 1 is a MAX IV task name defined in a SERVER statement.

[pri] - Parameter 2 is the priority to use if no priority was cataloged for the task, and the task is not established or prescheduled. The default is 255.

[lmf] - Parameter 3 is the global load module file for the task. The default is LM.

[maxvcs] - Parameter 4 is the maximum number of Virtual Circuits (VCs) that a task can have open at any time. The default is 4.

[options] - Parameters 5, 6, and 7 can be any combination of the keywords listed below. The default is no options.

[AMA] Activate on message arrival. If not specified, a resume occurs.

[SOA] Save task's open VCs and messages if the task aborts; otherwise close VCs and delete messages.

[SOE] Save task's VCs and messages if task exits; otherwise close VCs and delete messages.

NOTE: If the PORT statement is specified, the task-name must be specified. If any of the other parameters are missing, they are defaulted. If a PORT statement is not given for a task mentioned in a SERVER statement, all parameters for that task's port are defaulted. Standard defaults are:

Parameter			Default
2	pri	priority	255
3	lmf	load module file	LM
4	maxvcs	maximum number of VCs	4
5-7	options	options	none

EXAMPLES

PORT A
Define a port with standard defaults for task A.

PORT B,,,,AMA,SOE,SOA
Define a port for task B with standard defaults except add options AMA, SOE, and SOA.

PORT C,200,BM,100,AMA
Define a port for task C with activation priority 200, load module file BM, 100 Virtual Circuits, and the option to activate the port's task on message arrival.

DEFINE AN ITC SERVER

The SERVER statement is optional. It defines an ITC (Intertask Communication System) server name in the system being generated. The server created has meaning only if the INTERTASK macro is also used. More than one server name may refer to a given task.

SYNTAX

```
                                1           2            3           4
     SERVER           server-name task-name [userext] [userext]

                              5          6
                      [userext] [userext]
```

server-name - Parameter 1 is a string of from one to forty ASCII characters terminated by a comma. The character string must NOT include any blanks, control characters (non-printing), slash (/), equal (=), plus (+), minus (-), hexadecimal (#) or dollar ($) sign characters.

task-name - Parameter 2 is a MAX IV task name.

[userext] - Parameters 3 through 6 are optional words that can contain the use r extension of the server definition. If any of these words are omitted, a zero value is supplied.

NOTE: Each task defined by a SERVER statement in the SYSGEN may have a PORT statement defining task specific information. If no PORT statement is entered for a task defined in a SERVER statement, one is implicitly generated using the PORT statement defaults.

EXAMPLES

```
     SERVER  B_TASK,B
          Define  server  B_TASK and relate this server to task B.
          Set user data to Ø,Ø,Ø,Ø.

     SERVER  A_TASK,A,1,2,3,4
          Define a server A_TASK and relate this server to task A.
          Set user data to 1,2,3,4.

     SERVER  C_TASK,C,1,,,4
          Define  server  C_TASK and relate this server to task C.
          Set user data to 1,Ø,Ø,4.
```

ROCDIR

SPECIFY A RESIDENT OC DIRECTIVE

Specify one ROCDIR statement for each OC directive that is to be memory resident. An entry in the Resident OC Directory is made for the OC name. Object for the directive must be in the System Elements Library. Additionally any secondary overlays called by the directive must be specified by a ROCDIR statement and the overlay's object must also be on the System Elements Library. This statement is optional. Refer to Appendix I for the names of standard OC Directives.

SYNTAX

```
                   1
     ROCDIR       name
```

name - Parameter 1 specifies the name of an OC directive that is to be memory resident. It must be a 1- to 3-character CAN-code name.

EXAMPLES

```
     ROCDIR  ABO
          /ABO is a resident OC directive.

     ROCDIR  A
          /A is a resident OC directive.

     ROCDIR  E$M
          Error message secondary overlay is memory resident.
```

SPECIFY A PACKAGE OF OC DIRECTIVES

Specify one ROCPAC statement for each package of OC directives that are to be memory resident. An entry in the Resident OC Directory is made for each OC directive included in the package. Object for all directives and their secondary overlays must be in the System Elements Library. This statement is optional. Refer to Appendix I for the names of standard OC packages.

SYNTAX

```
                         1
     ROCPAC          packagename
```

packagename - Parameter 1 specifies the name of a package of OC directives that are to be memory resident. It must be a 1- to 3-character CAN-code name. Refer to Appendix I for the names of the OC packages available.

EXAMPLES

```
     ROCPAC  TAS
          TASK package of resident OC directives.

     ROCPAC  BIO
          BIOS package of resident OC directives.

     ROCPAC  ALL
          ALL resident OC packages.
```

MLSEQU

SELECT SEQUENTIAL TASK LOADER

The MLSEQU statement is optional. If the statement is not included in the SYSGEN, the system uses the standard MAX IV Module Loader. If MLSEQU is specified, the standard MAX IV Module Loader is replaced with the Sequential Task Loader.

If the MLSEQUENTIAL statement is included in the SYSGEN, there must be no prescheduled tasks or resident directories.

The system expects to find (on the logical file specified) a series of one or more modules in TOC GET format terminated by an End-of-File mark. All of the modules are sequentially loaded into the system task queue. The tasks can be started by using OC Commands or specifying PECULIAR AUTOSTART during cataloging.

SYNTAX

```
                    1
     MLSEQU[ENTIAL] logfile
```

logfile - Parameter 1 specifies the global logical file from which the various tasks are to be loaded. This file must be accessible at system startup and must include TOC GETS of all tasks that are run in this system.

DEFINE OPERATOR COMMUNICATIONS PASSWORD

The OCPASSWORD statement (optional) specifies a password for access to the Operator Communications task. The password can be modified by the /PAS OC directive. The password is a global password for all remote terminal users (using TMP).

If this statement is not present in the SYSGEN, no password is required by the OCP TMP Directive. In other words, not using the OCPASSWORD statement allows use of the OC task by all remote terminal users (using TMP).

SYNTAX

```
                            1
     OCPASS[WORD]    [password]
```

[password] - Parameter 1 is optional. It is a 1- to 8-byte password that must be entered before the Operator Communications task can be delivered to the requesting terminal user. If this parameter is not present, no remote terminal user is given the OC package by TMP. In other words, an OCPASSWORD statement without this parameter inhibits any use of the OC task by remote terminal users.

EXAMPLES

```
     OCPASSWORD    ////////
          No remote terminal can obtain access to the OC task.

     OCPASS    *PWRD*
          OC password is *PWRD*.

     OCPASSWORD
          No remote terminal can obtain access to the OC task.
```

CLOCK

DEFINE SYSTEM CLOCK AND TIMER

The basic system clock interrupt (level 6) repeats continuously at a 200 Hz rate and drives several types of system clocks directly. However, the Time-of-Day clock, task-scheduling timers, and delay timers are updated (using interrupt level #E) at a lower frequency, if desired. The CLOCK statement defines the basic resolution of each "tick" of these timers. Timers that are updated by level #E are called the "low" timers. The CLOCK statement is optional.

SYNTAX

```
                              1          2          3            4
     CLOCK                [divider] [CALENDAR] [DISPLAY] [UTILIZATION]
```

[divider] - Parameter 1 specifies the value of a divider count that the 200 Hz clock interrupt routine uses to avoid passing on all of its interrupts to the "low" clock interrupt routine that executes at level #E. This parameter divides the basic frequency by an integer value. For example, if the value of this parameter is 4, only 1 out of every 4 interrupts is passed to the level #E clock. If this count is one (1), every level 6 interrupt is passed on to level #E. The default value is 2, which provides a 0.01 second tick interval for all low clocks and timers. This parameter must be evenly divisible into 200.

[CALENDAR] - If Parameter 2 is specified, then a month/day table is attached to the data structure of the real-time clock. At midnight, the time-of-day clock updates a day, month, and year timer, using this table to determine the various peculiarities of day/month/year boundaries.

[DISPLAY] - If Parameter 3 is specified, then a memory display routine is added to the basic clock interrupt routine that updates a display of any memory cell (with a virtual address or actual page displacement that is in the sense switches in Register Block #E, Register #E). The map image table or relative actual page used to display the specified address is determined by an operator directive.

[UTILIZATION] - Parameter 4 causes the system to include those modules that update all the CPU utilization timers of tasks. These routines also display the CPU utilization Bar Graph kept in location 4.

EXAMPLES

```
    CLOCK  1
         0.005 Second tick resolution.

    CLOCK  2
         0.01 Second tick resolution.

    CLOCK  20
         0.1 Second tick resolution.

    CLOCK  200
         1.0 Second tick resolution.

    CLOCK  ,CAL,,UTI
         0.01  Second  tick  resolution,  calendar extension, and
         utilization.
```

SPECIFY HARDWARE IMPLEMENTATION OF OPTIONAL INSTRUCTION GROUPS

The presence of this optional statement declares that hardware is included in the configuration to support the specified operation-code group of instructions (as explained below). A particular "arithmetic exception" trap (hardware interrupt level #5) routine is called to process exceptional conditions detected by these hardware instructions. The absence of these statements for a particular group declares that hardware for that group does not exist. If useful, the support may be specified for a group even though hardware does not exist in the system.

If the system being generated must perform on various configurations, some of which include hardware implementation of the instruction group and some of which do not, then both the software simulation and hardware trap subroutines may be specified. When the hardware implementation is present, the software simulation is never accessed and only consumes memory space.

SYNTAX

```
                          1
     HARDWA[RE]        { 1X }
                       { 3X }
                       { 5X }
```

{1X}
{3X}
{5X}
- Parameter 1 defines an operation-code group of instructions. For example, the 3X group is the optional hardware Floating Point instruction group. If hardware is present for this instruction group, an internal name of U5$3X is generated during SYSGEN and the standard software simulation package in the System Elements Library for this instruction group is NOT linked into the system nucleus.

EXAMPLES

```
     HARDWARE  3X
          CPU has hardware floating point.

     HARDWARE  5X
          CPU has communications processor hardware option.
```

SPECIFY SIMULATIONS OF UNIMPLEMENTED INSTRUCTIONS

The SIMULATION statement (optional) defines the name of a privileged software simulation routine to be used to process traps of unimplemented instructions that are caused by the execution of certain unimplemented operation code groups. If this statement is not specified for a particular instruction group, any task program executing such an instruction either aborts with the reason code REX, or is directed to a "local" trap processing routine supplied explicitly by the task program. If this statement is specified for a particular group of instructions, the appropriate simulation is entered after the unimplemented instruction trap is cleared.

SYNTAX

```
                  1
SIMULA[TION]    { 1X }
                { 3X }
                { 5X }
```

{1X}
{3X}
{5X}

- Parameter 1 defines an operation code group of instructions to be used, unless specific optional hardware implementation has been included in the system configuration. Each group has 16 basic operation codes (X) numbered #Ø through #F. The 3X group is normally reserved for the optional Floating Point instruction group and the name of the simulation subroutine is U4$3X, generated by its external entry point. This simulation routine exists in the standard MAX IV System Elements Library and is linked to the system nucleus if called by the system generation statement. There are no standard simulation packages for MAX IV for simulation of 1X and 5X instruction groups.

EXAMPLES

```
SIMULA    1X
     Simulate special instructions.

SIMULATION    3X
     Simulate floating point instructions if no hardware
     present.
```

DEFINE STRUCTURE OF MAP POOL AND SCHEDULING ALGORITHM

The MAPS directive is optional. If not specified, the system is configured so that large MAPS (1 and 2) are available for assignment to large (or small) tasks and small maps (4, 5, 6, and 7) are available for assignment to small tasks. Large MAP 3 is not available for assignment since the small maps share hardware with large MAP 3.

SYNTAX

```
                            1
     MAPS               [ LARGE ]
                        [ FREE  ]
```

[LARGE] - If the keyword LARGE is chosen for Parameter 1, it indicates that MAPS 1, 2, and 3 are the only available maps (none of the small maps are available). All tasks, regardless of size, are assigned to one of these large maps.

[FREE] - If the keyword FREE is chosen for Parameter 1, all maps are available for assignment. Large MAP 3 and the small MAPS (4, 5, 6, and 7) are initially available in the map pool, with the following rules being applied when it is necessary to assign maps:

- o If a large map is needed MAP 3 is available, and all four of the small maps are available; then MAP 3 is allocated to the task and all four small maps are deleted from the pool. When MAP 3 is returned to the pool of available maps, all the four small maps are also returned to the pool.

- o If a small map is needed, one is available, and large MAP 3 is also available; the small map is allocated and MAP 3 is deleted from the pool. If at any time all four small maps are available in the pool simultaneously, MAP 3 is also returned to the pool.

Selection of the proper map allocation scheme is an installation dependent decision best made by the system manager -- perhaps after some experimentation with the particular application. The default mode is suggested as a starting point and should be suitable when there is a mixture of both large and small tasks executing asynchronously, particularly if the larger tasks are lower in priority than the smaller tasks.

EXAMPLES

```
MAPS FREE
     Specify FREE-FOR-ALL.

MAPS LARGE
     Specify LARGE maps only.
```

GLOBAL

DEFINE NAMES AND SIZES OF GLOBAL SHARED MEMORY AREAS

The GLOBAL statement allows blocks of actual memory, other than that already assigned to the resident nucleus of MAX IV, to be permanently allocated for global shared memory areas. A 6-character name can be assigned to each such area. Space is assigned where it can be found in the actual memory pool unless a specific actual page address is defined as the start of the area. In the latter case, the area may be positioned on the boundaries of memory pages shared by external CPU's or to boundaries that privileged tasks may agree upon for inter-communications. These memory areas may be designated as "external" (XCPU) so that the system does not diagnose or modify them during system startup.

SYNTAX

```
                        1    2      3          4         5        6
     GLOBAL           name pags [rdlock] [wrlock] [stpag] [option]
                                [rdinf ] [wrinf ]

                          7         8
                      [,option] [option]
```

name - Parameter 1 specifies the 1- to 6-character name of the shared memory area. This name may be referenced by programs that address the area (GLOBAL COMMON, ...) when the program is link-edited. Refer to the MAX IV LINK EDITOR (EDI) Programmer's Reference Manual, listed in the Preface.

pags - Parameter 2 defines the number of pages to be allocated to the area. These pages are allocated contiguously in actual memory so that privileged tasks may address them without necessarily mapping them into the task's virtual space(s).

If Parameter 6 is specified as XMEM, Parameter 2 specifies the number of 8K 16-bit word blocks in the extended memory region. Extended memory is supported for 16-bit tasks only.

[rdlock]
[rdinf] - Parameter 3 is an optional read-lock or read-influence limit. Any task that plans to read or execute code from the region must either supply the key to fit this lock or must have a cataloged influence limit that is compatible with the region. The influence limit must be supplied as negative. If nothing is supplied, the region is available to anyone for read and execute.

[wrlock]
[wrinf]

- Parameter 4 is an optional write-lock or write-influence limit. Any task that plans to write (or load) any information in the region must either supply the key to fit this lock or must have a cataloged influence limit that is compatible with the region. If nothing is supplied for this parameter the region is available to anyone for write purposes. The influence limit must be supplied as negative.

[stpag]

- Parameter 5 permits the defined memory area to be positioned at a specific starting point in actual memory. If not specified, the system attempts to resolve the required number of pages and assign them where it finds them.

  If Parameter 5 is specified, the memory region is diagnosed at start-up time (unless Parameter 6 specifies XCPU). If Parameter 5 is not specified, the resolved memory region is both cleared and diagnosed at start-up time.

  If Parameter 6 is specified as XCPU, this parameter is required. If Parameter 6 is specified as XMEM, this parameter is optional, but if specified must be on an 8K memory boundary.

[option]

- Parameters 6, 7, and 8 select optional characteristics of the defined region. The characteristics are selected by keyword and may be specified in any order. The keywords for the optional characteristics are:

[XCPU]

Used in conjunction with Parameter 5 to define a memory area shared by another "external" CPU. The area is put "off-limits" to all system routines that diagnose memory and compute memory size.

The following keywords, if specified, cause restrictions to be defined that override Parameters 3 and 4.

[READ[ONLY]]

Allows read-only access regardless of a task's ability to meet the write criteria.

[PRIV[ONLY]]

Allows insertion only by a privileged task.

[XMEM]

If XMEM is specified, READONLY should not be specified. An extended memory region is only accessible on a CLASSIC 7860/7870 computer.

EXAMPLES

```
GLOBAL  BABY,1,-127,-127
```

Defines a small global region of one page. Any task with an influence limit of at least 127 (Ø < X <= 127) is able to read or write freely. Any other task (127 <= X <= 255) cannot insert this global region.

```
GLOBAL  BIGJON,64,,BJ:,512,XCPU
```

Defines a large global region that anyone can insert and read or execute code in but only tasks that specify the key "BJ:" can write into it. The region starts at an actual page that is shared by two CPU's.

```
GLOBAL  A,2,,,#16Ø,XMEM
GLOBAL  B,1,,,#1AØ,XMEM
GLOBAL  AB,3,,,#16Ø,XMEM
```

Define extended memory regions of actual memory. Overlap regions A and B.

```
GLOBAL  C,1,,,,XMEM
```

Define an extended memory region of 8K 16-bit word blocks.

RESERVE SPACE FOR DIRECTORY OF PRIVATE SHARED MEMORY REGIONS

The SHPRIVATE statement (optional) defines the size of a directory (initially empty) that permits a maximum number of private shared memory regions to be created, inserted, erased, and destroyed during system operation. These regions are similar to GLOBAL shared regions but only exist as system definitions when at least one task is using them. A task can create such a region, activate a second task, let the second task insert the region in its addressing space and the original task can then disappear. The region can be passed from one task to another and is deleted only when no task is currently using it.

NOTE: If Parameter 2 (maxpages) is referenced on both the SHLOMO and SHPRIV statements, the value used is from the last encountered statement in the SYSGEN source file.

SYNTAX

```
                         1           2
    SHPRIV[ATE]      numregions [maxpages]
```

numregions - Parameter 1 defines the maximum number of private shared memory regions that can be defined at any one time.

[maxpages] - Parameter 2 sets the maximum number of memory pages from the actual memory pool that can be assigned to private shared regions. It is required either here or on the SHLOMO statements. These pages are not allocated until needed.

If maxpages is not large enough, then shared private regions will not be created. Refer to the SHLOMO statement in this manual.

EXAMPLES

SHPRIVATE 4,12
Specify four regions that may use a total of twelve pages (maximum) among them.

SHPRIV 2,32
Specify two regions that may use a total of 32 pages (maximum) among them.

SHLOMO

RESERVE SPACE FOR DIRECTORY OF SHARED LOAD MODULES

The SHLOMO statement (optional) defines the size of a directory (initially empty) that permits a maximum number of shared load modules to concurrently execute during system operation. An entry is made in the directory when a task loads a shared load module that is currently not in the directory. The module can then be shared by other tasks. The directory entry remains until no task in the system is using it.

NOTE: If Parameter 2 (maxpages) is referenced on both the SHLOMO and SHPRIV statements, the value used is from the last encountered statement in the SYSGEN source file.

SYNTAX

```
                           1           2
    SHLOMO            numentries [maxpages]
```

numentries - Parameter 1 defines the maximum number of shared load modules that can exist in the system any one time.

[maxpages] - Parameter 2 sets the maximum number of memory pages from the actual pool that can be assigned to private shared regions. It is required either here or on the SHPRIVATE statement.

If maxpages is not large enough, then shared load modules will not be shared, even though empty directory entries exist. No error is given by the system to indicate this condition. Refer to the SHPRIVATE statement in this manual.

EXAMPLES

SHLOMO 8
Srecify maximum of eight shared load modules.

SHLOMO 3,4Ø
Specify maximum of three shared load modules, plus forty total pages for private shared use.

RESERVE SPACE FOR RESIDENT LOAD MODULE FILE DIRECTORIES

The directory information for some load module files may be accessed so frequently that it is worthwhile to make their directories, or part of their directories, permanently resident in memory. When new cataloging activity occurs for the file, the Task/Overlay Cataloger updates both disc-resident and memory-resident directories if the cataloger is itself cataloged as a privileged overlay. The DIRECTORY statement is optional.

NOTE: This statement must not be used on a discless system.

SYNTAX

```
                     1        2
     DIRECT[ORY]     [fname] [entries]
```

[fname] - Parameter 1 is the logical file name (in the global file list) of a load module file. This file must be assigned to a disc file capable of containing quick-load module data. The default logical file name is LM.

[entries] - Parameter 2 is the number of entries allowed to be resident. This number of entries (starting at the beginning of the directory) is loaded into memory by the system's RESTART routine. When references are made to these files, these entries are checked first. If the requested program is not found in the resident directory, then a disc directory search is invoked by the loader. When programs are recataloged online, the resident directory is updated as well as the disc-resident copy. The default value is 25.

EXAMPLES

```
     DIRECTORY   LM,5Ø
          Specify file name of LM and fifty resident entries.

     DIRECT   BM,1Ø
          Specify file name of BM and ten resident entries.
```

PARITY

DEFINE PARITY TRAP PARAMETERS

This optional statement is provided to allow the user to change the default values that software parity error recovery uses on a CLASSIC 78xy computer.

SYNTAX

```
                          1          2                 3
     PARITY            [period] [[total-errors] [plane-errors]]
```

[period] - Sets integration time period for parity error rate timers on a CLASSIC 78xy computer specified in ticks. The default is 200 ticks.

[total-errors] - Maximum total number of errors that are logged during abo ve period. The default is 10 total errors.

[plane-errors] - Maximum number of errors that can occur per plane before error correction mode is automatically enabled for a plane. The default is 5 plane errors.

EXAMPLES

```
PARITY  10,3,2
```

Specifies period of 10 ticks, total errors three and plane errors two.

```
PARITY  ,20,10
```

Specifies default period of 200 ticks, total errors 20, and plane errors ten.

NOTE: For derivation of the tick value refer to the CLOCK SYSGEN statement in this manual.

DEFINE TARGET MACHINE

This optional statement is provided to allow the user to eliminate any machine dependent code from the system. If the statement is not entered, no code will be deleted.

SYNTAX

```
                    1
     MACHINE       MCV
```

MCV - Parameter 1 states that the system has a CLASSIC 78xy computer.

EXAMPLES

```
     MACHINE   MCV
          Specifies CLASSIC CPU.
```

SYSID

DEFINE SYSTEM IDENTIFICATION START-UP MESSAGE

The SYSID statement is optional. It generates a system start-up message consisting of from one to nine items of information. Each item appears in the start-up message with a space between it and the next item. (The operating system revision and date of SYSGEN are prefixed onto the user's system start-up message.)

SYNTAX

```
                          1      2       3         9
     SYSID              par1 [par2] [par3]...[par9]
```

par1 - Parameter 1 is the first word or string of characters that appears in the start-up message at system COLD START.

[par2]
[par3]
.
.
.
[par9] - Parameter 2 through 9 (optional) represent the second through ninth words or string of characters that appears in the start-up message.

EXAMPLES

```
     SYSID  [[[,THIS,SYSTEM,IS,A,DEVELOPMENT,SYSTEM,]]]

     SYSID  ENTER,TIME,AND,DATE
```

DEFINE A CPU-ID FOR THE SYSTEM

The CPUID statement is optional. It defines an ID for the system being generated. This ID may be used to uniquely define a system. The CPUID is accessible to user programs by REX service #4Ø.

SYNTAX

```
                        1        2
     CPUID          [num-1]  [num-2]
```

[num-1] - Parameter 1 is a single precision assembly value that is to be used as the system CPUID. Default is Ø.

[num-2] - Parameter 2 is a single precision assembly value and is used by MAXNET (see MAXNET user documentation for a description of this value). The default for this parameter is 2.

EXAMPLES

```
     CPUID  #1ØØ1
          Use ID #1ØØ1 for this system.
```

DEFINE A CPU-NAME FOR THE SYSTEM

The CPUNAME statement is optional. It defines a name for the system being generated. This name may be used to uniquely define a system. The CPUNAME is accessible to user programs by using REX service #40.

SYNTAX

```
                    1
CPUNAME          [string]
```

[string] - Parameter 1 is a string of from 1 to 64 characters (excluding non-printing characters and the comma (,), slash (/), plus (+), minus (-), hexadecimal (#), and dollar ($) sign). A blank is used to terminate the string. The default is a blank and a NUL. In all cases, 32 words are reserved.

EXAMPLES

```
CPUNAME   TEST_SYSTEM_NUMBER_1
```

DEFINE THE POOL OF TASK-SCHEDULING TIMERS

The TIMER statement (optional) defines one or more timers for use by the CONNECT service for scheduling (to ACTIVATE, RESUME, or KILL) a task. Repeated use of the statement causes the sum of all specified timers to be included in this pool of system resources. These timers are driven by the "low" clock frequency (at the tick rate) when allocated (CONNECTed) to a task and enabled (THAWed).

SYNTAX

```
                     1         2
     {TIMERS}        number [influence]
     {TIMER }
```

number - Parameter 1 specifies the number of timers to be generated by the statement. One or more can be specified for each usage of the statement. The default value is 1.

[influence] - Parameter 2 specifies the influence limit that tasks must have to be allocated this timer or any of this group of timers. Tasks having an influence limit that is incompatible (that is, a higher number) can be never allocated any of these timers, even when available. If not specified, a limit of 255 is used, allowing any task to allocated the timer(s) if it is available. Repeated use of the statement permits various timers to be defined with various limits (ensuring that a minimum number of timers are available that can never be allocated to "non-critical" tasks).

EXAMPLES

```
     TIMER
     TIMER
     TIMER
```

Defines three timers that are available for allocation to any task on a first-come/first-served basis.

```
     TIMER   6,64
```

Defines six timers that are available only to tasks with an influence limit between Ø and 64.

```
     TIMER   4,255
```

Defines four timers (as in the first example).

DCTIME

DEFINE DIRECTLY-CONNECTED TIMERS (FOR PRIVILEGED TASKS)

The system has several simple timers, which it manipulates directly with privileged routines, for purposes of timing certain system events. These timers consist of a simple linked list of 16-bit timers, with individual words that are updated directly by the system clock interrupt (at the basic 200Hz clock frequency), whenever they contain a negative number. Since the words in this list are manipulated directly, only a privileged task that can address MAP 0 can use them. The DCTIMERS statement (optional) permits this list to be expanded to include additional entries that can be manipulated by privileged user-coded tasks.

SYNTAX

```
                        1
     DCTIME[RS]     [number]
```

[number] - Parameter 1 specifies the number of additional timers to be added to the system's own timer list. A default value of one is used if the parameter is missing. A default value of zero is used if the DCTIMERS statement is missing.

EXAMPLES

```
     DCTIMERS  5
          Define five simple timers.

     DCTIME   2
          Define two simple timers.
```

DEFINE THE LIST OF DEDICATED TASK-SCHEDULING INTERRUPTS

The INTERRUPT statement (optional) must be repeated as necessary to define all interrupts that are to be dedicated as task schedulers. (Refer to the CONNECT service in the MAX IV Executive (REX) Services, System Guide listed in the Preface.) Only "external" levels (not already dedicated for system usage) may be specified.

SYNTAX

```
                     1        2           3
     INTERR[UPT]     level [influence] [entrypoint]
```

level - Parameter 1 specifies the interrupt level number to be dedicated as a task scheduler. It can be one of the following true "external" hardware priority levels, depending upon hardware configuration:

#7 #8 #9 #A #B

It can also be one of the following "pseudo-levels" that are provided by a special external interrupt card that can be connected to an IOIS chassis. These pseudo-levels actually utilize unused I/O party-line interrupt levels of true priority levels #C and #D.

Pseudo-Level	Actual Level	IOIS Chassis
#20-#27	#C	1st
#28-#2F	#C	2nd
#30-#37	#D	1st
#38-#3F	#D	2nd

[influence] - Parameter 2 specifies the influence limit that a task must have in order for it to be permitted to allocate this interrupt level. Tasks having influence limits incompatible with this number cannot allocate this interrupt level, even when available. If not specified, an influence limit of 255 is used, allowing any task to allocate the interrupt level if it is available.

[entrypoint] - Parameter 3 specifies a special entry point used if the interrupt level is to process custom instructions before passing control to the standard interrupt bootstrap program. The default is the entry point to the standard interrupt bootstrap program. If a user-defined entry point is specified, a subroutine by this name must be added to the System Elements Library and subsequently linked to the resident system nucleus. Methods of coding such a privileged subroutine are described in the MAX IV GENERAL OPERATING SYSTEM, System Guide Manual. If the same subroutine is used by more than one interrupt level, it must be coded in reentrant form.

EXAMPLES

INTERRUPT #Ø7,Ø25

Specify level #Ø7 as a task scheduling interrupt that only a task with influence limit between Ø and 25 can allocate.

INTERRUPT #27,15Ø,MYNTRE

Specify pseudo-level #27 as a task scheduling interrupt that only a task influence limit between Ø and 15Ø can allocate and, in addition, connect a user-supplied interrupt processing routine named "MYNTRE" that executes whenever this interrupt is processed.

INTERR #ØB,255

Specify level #ØB as a task scheduling interrupt that any task can allocate.

DEFINE THE "COMMON-ALARM" TASK-SCHEDULING INTERRUPTS

The CALARM statement (optional) permits the interrupt-multiplexing hardware of a common-alarm digital-input channel (IOIS option) to be supported as a task scheduler. The change-of-state of each individual bit of the 16-bit digital input unit may be defined as a pseudo-interrupt level and allocated by the CONNECT service to ACTIVATE, RESUME, or KILL any task. Only common alarm channels that cause an interrupt when each bit changes state are supported.

SYNTAX

```
                       1         2        3              4
     CALARM            #groupunit level [triggerbits] [influence]
```

#groupunit - Parameter 1 specifies the I/O group number and unit number (concatenated as one hexadecimal number "gu") where the common alarm input channel is plugged into the IOIS chassis. The first unit specified by a CALARM statement is assigned the pseudo-interrupt levels #4Ø-#4F. The next unit is assigned pseudo-levels #5Ø-#5F, and so on, up to a maximum of 12 common alarm channels. The least significant hexadecimal digit of the pseudo-level corresponds to the bit number (#Ø-#F) of the digital input channel.

level - Parameter 2 specifies the true hardware interrupt level, or IOIS pseudo-level, to be multiplexed and driven by the common alarm. Levels #Ø7 through #ØB and #2Ø-#3F are usable, but when specified, cannot also be specified with an INTERRUPT statement. (The common alarm provides for "trading-in" one interrupt level and gaining 16 in return.)

[triggerbits] - Parameter 3 allows the setting of trigger state bits (#ØØØØ-#FFFF) where:

Ø = Trigger on transition from 1 to Ø
1 = Trigger on transition from Ø to 1

If not specified the value #FFFF is assumed.

[influence] - Parameter 4, is optional. It specifies the influence limit a task must have to allocate any of the pseudo-interrupt levels (#xØ-#xF) defined by this common alarm channel. Tasks having influence limits incompatible with this number cannot allocate any of these pseudo-levels, even when available. The default value is 255.

EXAMPLES

CALARM #20,#0A,,255

Specify 16 pseudo-levels (#40-#4F) at common alarm digital input channel group #02 and unit #00 in the first IOIS chassis that uses true interrupt level #0A and can be allocated by any task as schedulers.

CALARM #21,#08,,100

Specify 16 pseudo-levels (#50-#5F) at common alarm digital input channel group #02 and unit #01 in the first IOIS chassis that uses true interrupt level #08 and can only be allocated by tasks with an influence limit less than or equal to 100.

DEFINE ALL DIRECTLY-CONNECTED INTERRUPTS (FOR PRIVILEGED USE)

The operating system contains many directly-connected interrupt levels that are permanently connected to privileged system subroutines. The DCINTERRUPT statement (optional) permits a system programmer to directly-connect a custom interrupt processing subroutine to unused external interrupt levels. Such routines must be included in, or ahead of, the MAX IV System Elements Library so that they can be linked to the system nucleus during the linking phase of system generation.

SYNTAX

```
                     1       2
    DCINTE[RRUPT]  level entrypoint
```

level - Parameter 1 specifies the external interrupt level (#7 through #B).

entrypoint - Parameter 2 specifies the 6-character external name of the entry point of the subroutine that must be linked to the system nucleus as a separately-assembled object module. Such a subroutine is only executed if enabled by a privileged task, custom executive service, or the system's RESTART Initialization routine.

EXAMPLES

```
DCINTERRUPT  #09,MYDCIN
```

Claim external interrupt #9 and connect user-supplied routine "MYDCIN" to it.

```
DCINTE   #23,MYPSEU
```

Claim pseudo-interrupt #23 and connect user-supplied routine "MYPSEU" to it.

In each example given above, the user must also supply a routine (invoked, perhaps, by system RESTART exits U$AINI, U$VINI, or U$XINI) to enable the respective interrupt level.

ENABLE/DISABLE THE TOCLOCK FEATURE

The TOCLOCK statement is optional. It is disabled if no TOCLOCK statement is provided or if TOCLOCK Ø is specified in the SYSGEN system structure block.

The TOCLOCK feature (when enabled) makes an effort to protect the integrity of load module files at two levels during activities that affect the load module directory. TOCLOCK guards against concurrent accesses of a load module file during directory manipulation. TOCLOCK also attempts to prevent the module loader from loading a module that is currently being manipulated by TOC.

NOTE: While all attempts have been made to "close all windows" during load module directory manipulation, there may be some sequence of events that can jeopardize file integrity.

TOCLOCK uses a data structure in MAP Ø to communicate between copies of TOC and between TOC and the module loader. The number of entries in the TOCLOCK data structure is determined by Parameter 1. If Parameter 1 is defaulted in the SYSGEN source, TOCLOCK is enabled and a four entry data structure is created.

A system that performs several load module file updates simultaneously, or a system that performs load module file updates while running shared or segmented programs may need more than four entries. The TOCLOCK data structure size can be adjusted by Parameter 1.

A realtime system that is MAP Ø tight and never does online load module directory manipulation can remove the TOCLOCK data structure and REX service from MAP Ø by specifying TOCLOCK Ø or by removing the TOCLOCK statement from the SYSGEN.

NOTE: No TOCLOCK statement or TOCLOCK Ø removes system protection from all load module file directory manipulation.

If the user simply has TOCLOCK in the SYSGEN, the number of TOL entries is 4.

If the user specifies TOCLOCK Ø or has **no** TOCLOCK statement in the SYSGEN, there is **no** TOL structure. A TOCLOCK Ø statement disables the TOCLOCK feature and eliminates the data strucure and REX service from MAP Ø.

The user can specify the number of TOCLOCK entries by specifying the number along with the TOCLOCK statement in the SYSGEN. Example: "TOCLOCK x" creates x TOL entries.

SYNTAX

```
                         1
    TOCLOC[K]          [number]
```

[number] - Parameter 1 (optional) specifies the number of elements in the TOCLOCK list. (Default = 4)

If parameter 1 is specified as Ø, the TOCLOCK feature is disabled.

EXAMPLE

TOCLOCK Ø

Disable the TOCLOCK feature. Eliminate the data structures and REX service from MAP Ø. Remove security during load module directory manipulation.

DEFINE SYSTEM DEBUG PARAMETERS

The SYSBUG statement is optional. It specifies both the size of the histogram stack used by the SHADOW OC directive, and whether the stand-alone dump package is to be included in the System nucleus. To use the PAUSE OC directive, this statement must be specified with a non-zero histogram stack size. If the ADUMP option or BOTH option on the SYSBUG statement is specified, this statement creates the Upper Control Block extension for the actual dump data structure.

SYNTAX

```
                         1       2       3
     SYSBUG          {words} {DUMP } [name]
                             {ADUMP}
                             {BOTH }
```

words - Parameter 1 is optional and indicates the size of the histogram stack, in words, to be reserved by the system. If zero is entered, the histogram stack is not installed in the system. The default value is a minimum of 17 words, which is the maximum number of values saved at one interrupt by the SHADOW OC Directive (two words for the PSD and 15 words for the register file).

{DUMP}
{ADUMP}
{BOTH} - Parameter 2 can be any one of the following:

DUMP, the default, causes the inclusion of the Virtual Dump Routine (VD.UMP). The starting address VD$UMP is in MAP Ø at location #4Ø1 preceded by a 'BRU' instruction so that loading a #4ØØ into the program register and setting RUN invokes the hexadecimal dump routine regardless of it's link-edited location.

ADUMP causes the inclusion of the actual memory dump package. A UCB extension (UCBADU) is created, a startup routine (AD.INI), and a standalone dump routine (AD.UMP) are included. The dump routine's address AD$UMP is stored at MAP Ø location #4Ø1.

BOTH causes both VD.UMP and AD.UMP to be included in the system. It brings in a routine called BD.UMP that checks the front panel switches to determine whether to use the VDUMP routine or the AD.UMP routine. The starting address BD$UMP is stored in MAP Ø at location #4Ø1.

[name] - Parameter 3 is a 1- to 3-character CAN-code name of the partition or magnetic tape unit that receives the memory image dump. The default value isSAV. This parameter is used when ADUMP or BOTH is specified for Parameter 2.

EXAMPLES

```
     SYSBUG  17Ø
          17Ø words are reserved, VD.UMP is not included.

     SYSBUG  Ø,DUMP
          No words are reserved, VD.UMP is included.

     SYSBUG  3,DUMP
          17 words are reserved, VD.UMP is included.

     SYSBUG  64,ADUMP,MT1
          64  words  are  reserved,  AD.UMP  is included.  Actual
          memory will dump to MT1.

     SYSBUG  64,BOTH
          64  words  are  reserved,  both  AD.UMP  and  VD.UMP are
          included.  Actual memory will dump to SAV.
```

DEFINE SYSTEM OPTIONS

The SYSOPT statement is optional. It generates a 16-bit data word. This option register is reserved for standard system related functions that alter the reactions of the system.

SYNTAX

```
                        1     2     3
    SYSOPT            [MSL] [DTS] [PWR]
```

[MSL] - Parameter 1 (optional), if present in a system with no ROLLER, causes a task requesting memory to go into memory stall (relinquish) if there is a task higher in the CPU queue awaiting memory.

[DTS] - Parameter 2 (optional) enables the date/time stamp capability for system messages. When enabled, all abort messages print a date/time stamp. Additionally, any user message can have a date/time stamp. This affects REX ABORT, KILL, MESSAGE, and WAIT.

[PWR] - Parameter 3 (optional) causes the Level Ø Interrupt Handler to increase to handle total power failure and recovery interrupts, AC lowline, AC recovery, and remote memory power failure and recovery.

EXAMPLE

```
    SYSOPT   MSL
        Memory stall until higher priority tasks get their
        memory.
```

EVENTL

SPECIFY THE EXISTENCE OF EVENT LOGGING

The EVENTLOG statement is optional. It adds the Event Logging mechanism to the system being generated. When this statement is specified, the following parameters configure the initial characteristics of the Event Logging mechanism. All parameters are optional.

SYNTAX

```
     EVENTL[OG]       1     2     3     4     5     6     7
                    [ENA] [ALT] [RES] [NOT] [lfn] [lfn] [lfn]
                    [DIS] [CYC] [SUS] [DON]

                      8      9      10     11     12    13
                    [num] [tname] [lfn] [size] [NOF] [lfn]
                                               [FOR]
```

[ENA]
[DIS]
- Parameter 1 specifies whether Event Logging is to be enabled (ENA) or disabled (DIS) following system startup. The default is ENA.

[ALT]
[CYC]
- Parameter 2 specifies the manner in which data records are stored on the Event Logging data files; either alternating (ALT) mode or cyclic (CYC) mode. The default value is ALT.

[RES]
[SUS]
- Parameter 3 specifies what execution state the Event Logging mechanism uses when a logging data file is full; either resume (RES) execution or suspend (SUS) execution. The default value is RES.

[NOT]
[DON]
- Parameter 4 specifies whether the operator should be notified (NOT) or not notified (DON) of Event Logging activities. The default value is NOT.

[lfn]
- Parameter 5 specifies the global logical file name of the Event Logging cyclic data file or the first of the two alternating data files. The default value of this parameter is SLF.

[lfn]
- Parameter 6 specifies the global logical file name of the second of the two alternating data files. The default value of this parameter is a ALF.

[lfn] - Parameter 7 specifies the logical file name of the Event Logging format file. The default value is LFF.

[num] - Parameter 8 specifies the number of logging nodes dedicated to the Event Logging mechanism by MAX IV. The default value is 2Ø.

[tname] - Parameter 9 defines the name of the Event Logging Report task. The default value is EVEREP.

[lfn] - Parameter 1Ø defines the global logical file name where the Logging Rep ort task is located. The default value for this parameter is SM.

[size] - Parameter 11 specifies the number of entry-specific data words for e ach logging node. The default value is defined by the size needed for the standard system logpoints. Refer to the LOGPOINT SYSGEN statement.

[NOF]
[FOR] - Parameter 12 specifies whether or not the event logging console message formatting routines are (FOR) or are not (NOF) to be included in the system. The default is NOF.

[lfn] - Parameter 13 specifies the global logical file name to be used for outp ut of the event logging formatted console messages. If not specified and the formatting routines are configured then the output goes to the global CO file.

EXAMPLES (only one permitted per assembly)

```
EVENTLOGGING
     Default values

EVENTL   ,,,,XLG,YLG,ZLG,,,,
     Specify files but default all other parameters.

EVENTLOG  ,,,,,,,,REPORT,LM
     Specify REPORT task on LM file, default all other
     parameters.
```

LOGPOI

SPECIFY EVENTS TO BE LOGGED

The LOGPOINT statement is optional. If specified it must be used in conjunction with and following the EVENTLOG statement. The LOGPOINT statement specifies the specific events in the MAX IV system that are captured and logged by the Event Logging mechanism.

SYNTAX

```
                        1        9
     LOGPOI[NT]       [point]...[point]
```

[point] - All points are optional. They are selected by keyword and may be specified in any order. Each point specified results in the logging of the particular event type. The keywords for the [point] argument are:

ACT[IVATE] Monitor and log all task activations with the exception of the system OC task.

EXI[T] Monitor and log all task exits with the exception of the OC task.

ABO[RT] Monitor and log all task aborts with the exception of the OC task.

INT[ERRUPTS] Monitor and log all unsolicited hardware interrupts.

PAR[ITYERRORS] Monitor and log all memory parity errors.

BAD[PAGE] Monitor and log all unrecoverable memory parity errors that result in removing a memory page from an allocatable state.

RAT[E] Monitor and log excessive memory parity error rates that result in the enabling of hardware parity error correction.

OFF[LINE] Monitor and log all device offline conditions.

RET Monitor and log all successful I/O retry operations.

ALL Monitor and log all of the above.

EXAMPLES

```
LOGPOINT  ACT,EXI,ABO
     Monitor and log all task activation, exits, and aborts.

LOGPOI   PARITYERROR,BAD
     Monitor and log all memory parity errors and
     unrecoverable memory errors.
```

LOGFOR

SPECIFY EVENTS TO BE FORMATTED TO THE CONSOLE

The LOGFORMAT statement is optional and, if specified, must be used in conjunction with the EVENTLOG statement. Parameter 12 of EVENTLOG must specify FORMAT. The LOGFORMAT statement is used to specify which events are to be formatted to the formatted message file as they occur. An event must be captured before it is formatted. Refer to the LOGPOINT SYSGEN statement.

SYNTAX

```
                     1        9
    LOGFOR[MAT]    [point]...[point]
```

[point] - All points specified are optional and can be specified in any order. Each point specified results in the immediate formatting of that event if logging, formatting, and logging of that event are all enabled. Possible values for this argument are the same as those for the LOGPOINT SYSGEN statement. Parameter 1 defaults to ALL.

EXAMPLES

```
    LOGFORMAT
        Format all standard events.

    LOGFOR   EXI
        Format only task exits.
```

SPECIFY PROGRAMMER'S WORKBENCH WANTED

The WORKBENCH statement is optional. It includes the Programmer's Workbench REXs (REX #ØX22) and the UCB extension for the Workbench in the system. This statement may appear anywhere in the PART 1 source. A SERVICE call is not made for Workbench when a SERPAC is specified.

SYNTAX

```
                     1
     WORKBE[NCH]    [count]
```

[count] - Maximum number of spawned tasks, generated by one terminal, that can be in the CPU queue at one time. Default is 1Ø tasks.

EXAMPLES

WORKBENCH

Include the Programmer's Workbench REXs and permit a maximum of 1Ø spawned tasks generated by one terminal to be in the CPU queue at one time.

WORKBE 5

Include the Programmer's Workbench REXs and permit a maximum of 5 spawned tasks generated by one terminal to be in the CPU queue at one time.

DECLARE END OF FIRST SYSTEM GENERATION ASSEMBLY

The ENDSYSTEM statement is required. It finalizes the basic system data structures and terminates the Macro Assembly (outputs an END statement) for this first part of the system generation source file. The resulting "incomplete" main program defines one-third (or one-fourth) of the standard system data structures and must be followed by other Macro Assemblies that are defined in PART 2 and PART 3 (and optionally, PART 4).

SYNTAX

```
ENDSYS[TEM]
```

EXAMPLES (only one usage permitted per assembly)

```
ENDSYSTEM

ENDSYS
```

2.2 PART 2 - BASIC I/O SYSTEM BLOCK

Use of the statements described in PART 2 should immediately follow those statements selected from PART 1. A separate Macro Assembly is performed on each part so that the system generation process can operate with a smaller batch-processing task memory requirement. That is, three separate macro assemblies require less memory space for macro storage and symbol table storage. The PART 2 Macro Assembly constructs several data structures used only by the Basic I/O System, such as a threaded list for each of the following:

- All timed I/O device controllers
- All untimed I/O device controllers
- All symbiont "controllers"
- All device transports
- All program addressable I/O devices (including disc devices, non-disc devices, and imaginary devices (simulated by symbionts))

LISTING TITLE - BEGIN SECOND ASSEMBLY

The TTL standard Macro Assembly statement is optional and can contain any message. Its purpose is to ensure that the SYSGEN listing starts on a whole page boundary --- and that the title specified appears on each page of the listing.

SYNTAX

```
                       1
     TTL             heading
```

heading - Parameter 1 contains the heading that appears at the top of each page of this part of the SYSGEN listing.

EXAMPLE

```
     TTL   MAX IV: I/OBLK
```

INSERT MACRO PROTOTYPES - TO DEFINE MACRO LANGUAGE

Before the macro language that defines the MAX IV input/output structures can be used, the prototypes that define the language must be inserted from an external file. This standard Macro Language statement performs this function and is required.

SYNTAX

```
                           1      2
     INS                 fname  IVIOS
```

fname - Parameter 1 specifies the logical file name where the macro prototypes are cataloged. The MAX IV standard for this file is MC, the Common Macro Library. This file name must be assigned to a real file or device before it can be used.

IVIOS - Parameter 2 is the name of the particular macro-group used to define the statements in the Basic I/O System Block.

EXAMPLE (only one usage permitted per assembly)

```
     INS    MC,IVIOS
```

All the SYSGEN statements in PART 2 are defined by the MAX IV Basic I/O System Structure Prototypes. These definitions are included in the assembly by using this statement.

NAME OF PROGRAM

This standard Macro Assembler statement is optional and assigns a name to the program being assembled. This statement must follow the INS statement in order to override the standard default provided by the macro file inserted for this block of the SYSGEN.

SYNTAX

```
                    1
      PGM          name
```

name - Parameter 1 is a valid symbolic name as defined in the MODCOMP Assemblers, Language Reference Manual. GI.IV is the default name for the Basic I/O System Block.

DEFINE EACH HARDWARE CONTROLLER CHARACTERISTICS: GROUP/UNIT/DMP...

The CONTROLLER statement is required for each timed or untimed hardware I/O controller used in the configuration. A unique name must be given to each defined hardware controller so that it can be referenced by name when individual devices (such as transports, packs, or units) of a controller are defined later.

The data structure produced by this statement is a data table that the Basic I/O System uses as a final target for all I/O operations queued for the devices that share the controller. Since the controller (the lowest hardware level) is the final "serializer" of I/O requests, it is at the controller table that the I/O queue of all currently scheduled operations is maintained.

A set of unique interrupt-receiving instructions is also generated by the statement, which are directly-connected to the appropriate unique data/service interrupt vectors for the channel. These interrupt receiving instructions also transfer control to a handler that may be common to several controller channels. This Physical Device Table (PDT) containing the I/O queue, the group/unit, and DMP assignment is passed to the handler's common service and data (End-of-Block) interrupt processing routines. The table also contains some "scratch" cells that are used by the handler to permit it to be channel independent and reentrant.

Use of the CONTROLLER statement for a custom handler requires that the following statements be added to the SYSGEN source prior to the CONTROLLER statement. It is the user's responsibility to provide the correct value depending on the device being added. Refer to values listed in the macros. These statements define default words for the following values in the data structures:

```
TCWX___   SET   Ø     TRANSPORT CHARACTERISTICS WORD
OPTX___   SET   Ø     LOGICAL DEVICE OPTION WORD
WDOG___   SET   Ø     WATCHDOG TIMER VALUE
CCWX___   SET   Ø     CONTROLLER CHARACTERISTICS WORD
POLV___   SET   Ø     POLLER TIMER VALUE
```

(___ is the controller name.)

SYNTAX

```
                        1       2     3       4           5      6
    CONTRO[LLER]  cname    #gu  [dmp1]  [UNTIMED]  [#gu2]  [dmp2]
                                [TIMED]

                            7        8        9       1Ø     11    12
                      [xbsiz]   [CPU]  [TABLE]  [cnm]  [PRI]  [ctv]
                                [bmeta]         [xcw]
```

cname - Parameter 1 is the controller name (generally given as "hhx") and must be unique to all controller names and symbiont task (controller) names. The first two characters (hh) uniquely select a standard or user-coded device handler that is later linked to the system during the linking phase. Include the third character (x) only as necessary to make all controller names unique that share a common handler (for example, TT, TT1, TTA, ... are controller names that share a common teletype (TT) handler).

#gu - Parameter 2 is the controller's group number (g) and unit number (u) concatenated into one hexadecimal quantity. Hexadecimal representation is the only practical way of specifying the controller address, except when the group number is zero.

[dmp1] - Parameter 3 is the optional number of the Direct Memory Processor (DMP) channel. When a device does not use a DMP, this parameter should not be specified. Valid numbers are #00-#0F for the first bus. Depending on the computer configuration the following numbers are valid:

#10-#1F for the second bus.
#20-#2F for the third bus.
#30-#3F for the fourth bus.

[UNTIMED]
[TIMED] - Parameter 4 is an optional keyword that determines whether this controller belongs to the list of untimed controllers or to the list of timed controllers. Timed controllers are examined by each clock interrupt. After the handler initiates a device operation on the controller, the clock terminates any operations that exceed a time limit set by the handler. Default is UNTIMED.

[gu2] - Parameter 5 is optional. It allows a special device controller to have a second controller address (group/unit). The only standard devices that use this feature are the analog input multiplexors that use separate I/O channels for sending addresses and receiving data.

[dmp2] - Parameter 6 is optional. It allows a special device controller to have a second DMP channel. The only standard devices that use this feature are the analog input multiplexors.

[xbsiz] - Parameter 7 is optional. It allows the expansion of the PDT by the number of words specified (all of which contain zeros). This parameter may be specified for any controller but the standard handlers do not use the extra words. User-provided handlers may require the

extra space and no restrictions exist on their use. A PDT extension containing non-zero words may also be added by reverting to Assembly language following the statement call.

[CPU] - Parameter 8 is optional. It specifies that this controller (probably a serial or parallel computer-computer link) is being used as a MAXNET device. It causes generation of additional words required by the MAXNET software.

[TABLE]
[bmeta] - Parameter 9 is optional and is only used by the Bulk Memory Controller. It specifies either the keyword "TABLE" or a decimal track address. Either parameter implies the desire to include the device-resident End-of-File (EOF) table. If the parameter is TABLE, then the location of the table in the Bulk Memory is defaulted to a track following the last logical device declared with a DISCDEVICE statement. This is the recommended position.

If the parameter is a decimal number, then the location of the table in the Bulk Memory will be at that particular track number. The default value is no device-resident EOF table (and an EOF can only be detected by reading the contents of a sector).

[cnm]
[xcw] - Parameter 10 is used for Model 4828 controllers. "cnm" is used for MAXNET links and "xcw" is used for non-MAXNET links.

cnm - Specifies the controller name mate for the MAXNET Link. The name must be a 3-character name, the first 2 letters of which must be SM. The third character makes it unique.

xcw - Specifies the extended characteristics word for a non-MAXNET link. It is used to specify the length of the combined address and control field and to specify operational flags.

[PRI] - Parameter 11 is a keyword used only for the Model 4828 MAXNET Standard Link and is used to distinguish between Primary and Secondary stations. The default value is secondary.

[ctv] - Parameter 12 is used only for the Model 4828 MAXNET Standard Link and is a cycle timer value expressed as a positive number of ticks. The default is 200.

EXAMPLES

```
     CONTROLLER TT,#ØB,,TIMED
          The first Teletype controller (console).

     CONTRO   CR,#Ø5
          The first card reader controller.

     CONTRO   MT,#Ø4,#Ø4
          The first magnetic tape controller.

     CONTRO   BM,#Ø2,#Ø2,TIMED,,,,,TABLE
          The first bulk memory controller with EOF table.
```

DEFINE EACH "CONTROLLER" FOR SYMBIONT SIMULATED DEVICES

The SYMCONTROLLER statement must be specified for every symbiont task (such as a spooler or a device simulator) to be installed in the system. It produces a data structure that is similar to the CONTROLLER data structure described for hardware I/O device channels. It is a structure that is the target for all operations that the Basic I/O System queues up for those "imaginary devices" simulated by a symbiont task. Each Symbiont Controller must be given a unique name so that it can be referenced by all the "imaginary" I/O devices to be defined subsequently (Refer to DEVICE). The main purpose of a symbiont having a permanently resident data structure is so that the symbiont task body may be non-resident and yet there will be a resident target for BIOS to queue to at all times. BIOS activates the symbiont task if it finds that it is non-resident. Notice that the symbiont task must be declared as PRESCHEDULED in the third SYSGEN assembly phase.

SYNTAX

```
                    1       2              3                  4
   SYMCON[TROLLER]  tname [xbsiz] [task-lvl-entry] [test-assign]
```

tname - Parameter 1 is the name of a resident or non-resident symbiont task (1- to 3-characters) that services (or simulates) the imaginary devices that reference this task (controller) name. Since this task becomes the controller, the name must be unique to all defined controllers.

[xbsiz] - Parameter 2 is optional. It allows the data structure generated to be extended beyond the standard size by the specified number of words as may be required for a particular symbiont task. Preinitialized words may also be added by simply reverting to Assembly language following the statement call. The standard MAX IV Printer Spooler (S task) requires some of these words to be configured if the spooler task is installed as a permanently-resident task.

[task-lvl-entry] - Parameter 3 is optional and allows the user to define a task-level entry point to the symbiont. This parameter defines the 3-character controller suffix to be applied to the prefix "TL$" to define the task-level entry point of a resident routine. The default value is: no task-level entry point is defined for this symbiont.

[test-assign] - Parameter 4 is optional and allows the user to define a test-assign entry point to the symbiont. This parameter defines the 3-character controller suffix to be applied to the prefix "TA$" to define the test-assign entry point of a resident routine. If this parameter is missing, no test-assign entry is defined for this symbiont.

EXAMPLES

```
SYMCONTROLLER  S
     Specify the output spooler task.

SYMCONTROLLER  ABC,128
     Specify   user-coded   symbiont  task  with  a  128-word
     extension buffer.

SYMCON   JKL
DFC      -1,#FFEØ,Ø
RES      256,Ø
     Specify  user-coded  symbiont task with three extra data
     words and a 256-word extension buffer.
```

NOTE: The symbionts specified above must also be declared as either resident (see TASK) or non-resident, pre-scheduled tasks (see PRESCHEDULE) in the third SYSGEN assembly (IVTASK).

DEFINE EACH COMMUNICATIONS CONTROLLER CHANNEL

The COMCONTROLLER statement must be specified at least once for each communications line used in a MAX IV system. All controller names specified across CONTROLLER, SYMCONTROLLER, and COMCONTROLLER statements must be unique. This statement generates the same data structure as that produced by the CONTROLLER statement with extensions. In addition, for each group of statements specifying the same group and unit number, a multiplexor table entry and appropriate driver code is produced. The statements specifying the same group and unit number must not be mixed with others specifying a different group and unit number. External references are also produced to ensure inclusion of the appropriate device handler during the link-edit phase.

SYNTAX

```
                       1     2    3       4           5     6     7
COMCON[TROLLER]      cname #gu  chnl  [UNT[IMED]]  [fdc] [fdn] [cls]
                                      [TIM[ED]   ]
                                      [VAC[ANT]  ]

                        8     9    10     11        12
                      [NET] [CL] [ssa] [parity] [baud]
```

cname - Parameter 1 is the communications controller channel name (of the form "hxx") and must be unique for all controller names. The following format is recommended:

h = Use A for an asynchronous channel, S for a synchronous channel, and M for a MAXNET channel.
xx = Use any CAN-code pair necessary for uniqueness.

When this statement is used to configure the Asynchronous Floppy Disc, the first two characters must be AF and only the third character is used for uniqueness.

#gu - Parameter 2 is the "even" group and unit number concatenated into one hexadecimal quantity for this controller.

chnl - Parameter 3 is the 3-character controller channel number. For half duplex, a COMCONTROLLER statement is required for each even numbered channel; for full-duplex, separate COMCONTROLLER statements are required for each output (even) and input (odd) channel.

[UNTIMED]
[TIMED]
[VACANT] - Parameter 4 is an optional keyword that determines whether this controller is timed. All synchronous controllers MUST be timed. The default value for this parameter is UNTIMED. VACANT specifies that this channel is skipped. This parameter leaves a hole in the multiplexor table. If VACANT is specified, parameters 1, 2, 3, 4 and 7 are the only parameters accepted.

[fdc] - Parameter 5 is used only for full duplex channels and specifies the other channel number of the channel to be used later, in conjunction with this one, to configure a full duplex channel.

[fdn] - Parameter 6 is used for only full duplex channels and specifies the controller name, as specified in Parameter 1, of the "mate" (see description of "fdc").

[cls] - Parameter 7 specifies the controller class. Enter one of the following keywords to define the class of the controller:

A = 4810, 4811, 4812, 4813, 1905 - Sync/Async
B = 4815, 4817 - Sync
C = 1906 - Sync/Async
D = 4804, 4808, 4809, 4858, 1907, 1907A, 1907B,
    1908 (DI) - Sync/Async
E = 1907, 1907A, 1907B (DMI) - Bisync
F = 1907, 1907A, 1907B, 1908 (DMI) - Sync/Async
G = 1907, 1907A, 1907B, 1908 (DMI) - HDLC/SDLC
H = 1907, 1907A, 1907B, 1908 (DI) - HDLC/SDLC
I = 4806, 4807 - Async
J = 4828 - MAXNET Std Link
K = 4804, 4808, 4809, 4858, 1907, 1907A, 1907B,
    1908 (DI) - AFD
L = 4806, 4807 - AFD
M = 1907, 1907A, 1907B, 1908 (DMI) - AFD
N = MODACS II

[NET] - Parameter 8 is a keyword used to specify that this channel will be used as a CPU link by the MAXNET symbiont.

[CL] - Parameter 9 is an optional keyword used to specify that an asynchronous device has a current loop interface and is required for 1907 asynchronous channels.

[ssa] - Parameter 10 is used for the Model 4828 MAXNET Standard Link only and is mandatory for such a device. It is the 3-digit secondary station address and must also be less than or equal to 255.

[parity] - Parameter 11 is a keyword specifying the parity selection for contro ller classes K, L, or M (Asynchronous Floppy Disc devices) and may have the following values:

EVEN	Even parity
ODD	Odd parity
NONE	No parity (default)

[baud] - Parameter 12 specifies the line speed for controller classes K, L, or M and may be specified as one of the following baud rates:

75, 11Ø, 134 (for 134.5), 15Ø, 3ØØ, 6ØØ, 12ØØ, 18ØØ, 2ØØØ, 24ØØ, 36ØØ, 48ØØ, 72ØØ, 96ØØ, 19.2 (indicating 19.2K baud), or the keyword EXTD for Extended/External (where the timing is supplied externally). The default value is 12ØØ.

EXAMPLES

```
COMCONTROLLER  AØØ,#18,ØØØ,,,,A
```

Asynchronous half-duplex channel on a 481Ø, 4811, or 19Ø5.

```
COMCON   AØØ,#1E,ØØØ,,ØØ1,AØ1,A
COMCON   AØ1,#1E,ØØ1,,ØØØ,AØØ,A
```

Asynchronous full-duplex channel on a 481Ø, 4811, or 19Ø5.

```
COMCON   SØØ,#1A,ØØØ,TIMED,ØØ1,SØ1,B
COMCON   SØ1,#1A,ØØ1,TIMED,ØØØ,SØØ,B
```

Full duplex channel on a 4815 or 4817 (synchronous) controller.

DEFINE THE CHARACTERISTICS OF A NON-STANDARD DISC CONTROLLER

The DISCCHARACTER statement is optional and is only required if a non-standard disc controller is configured in the system. If specified, it must follow the associated CONTROLLER statement and precede the associated FMTRANSPORT statement. Parameters 3 through 7 have standard system-supplied default values as long as the first characters of the controller name matches one of the standard controller names. Appendix C lists the default values.

SYNTAX

```
                           1     2      3     4     5     6
   DISCCH[ARACTER]     tname cname [bps] [spt] [tpc] [cpt]

                           7     8     9     10
                       [CYL] [EXC] [MOV] [xcw]
                       [NOC] [NOE] [FIX]
                                   [FLO]
                                   [BUL]
                                   [LAR]
```

tname - Parameter 1 is the transport name to which these characteristics are applied. This is the same name as used in the FMTRANSPORT statement.

cname - Parameter 2 is the name of the controller to which this transport is connected (Refer to CONTROLLER statement).

[bps] - Parameter 3 is optional. It specifies the number of bytes-per-sector for this disc transport on this controller.

[spt] - Parameter 4 is optional. It specifies the number of sectors-per-track (or sectors-per-revolution) for this disc transport on this controller.

[tpc] - Parameter 5 is optional. It specifies the number of tracks-per-cylinder (or tracks-per-radius) for this disc transport on this controller.

[cpt] - Parameter 6 is optional. It specifies the number of cylinders-per-transport (or cylinders-per-pack) for this disc transport on this controller.

GENERATES A MAP Ø RESIDENT PDT AND TCT FOR BLOCK MUX CONTROLLER

The MUXCONTROLLER statements must be specified once for each Block MUX Controller in the system. This statement must be specified in the Basic I/O System Block and should appear only after all of the COMCONTROLLER statements. MUXCONTROLLER generates a MAP Ø resident PDT and TCT for the Block MUX Controller. External references are also produced to include the Block MUX Handler during the link edit phase.

SYNTAX

```
                         1     2    3     4       5
   MUXCON[TROLLER]   cname,#gu,dmpl[,wdog][,link]
```

cname - Parameter 1 is the controller name and must be unique for all controller names

#gu - Parameter 2 is the controller's group and unit number concatenated into one hexadecimal quantity for this controller.

dmpl - Parameter 3 is the number of the Direct Memory Processor (DMP) channel. Depending on the computer configuration, the following numbers are valid:

#ØØ - #ØF for the first bus
#1Ø - #1F for the second bus
#2Ø - #2F for the third bus
#3Ø - #3F for the fourth bus

[wdog] - Parameter 4 is optional. It specifies the maximum time in two hundredths of a second that the Block MUX Handler waits for transfer of a message to complete. The default value for this parameter is 5Ø (1/4 second).

[link] - Parameter 5 is optional and specifies the total number of MAXNET links configured for this Block MUX Controller.

[CYL]
[NOC]

- Parameter 7 is optional. It specifies whether or not the controller provides automatic cylinder advance when overflow of the track address occurs. If the keyword CYL is given, then the handler assumes that cylinder overflow is handled by the controller. If the keyword NOC is given, then it is the responsibility of the handler to provide for cylinder overflow. The default for this parameter is based on the type of the controller, which is determined by the controller name.

[EXC]
[NOE]

- Parameter 8 is optional. It allows exclusive use of the entire transport. If the keyword EXC is given, exclusive use is allowed. If the keyword NOE is given, exclusive use is not permitted. The default is EXC.

[MOV]
[FIX]
[FLO]
[BUL]
[LAR]

- Parameter 9 is optional. It specifies the type of basic handler mechanism required for this transport. The default for this parameter is based on the type of controller, which is determined by the controller name.

  If the keyword MOV is given, this transport is handled as a moving head disc.

  If the keyword FIX is given, this transport is handled as a fixed head disc.

  If the keyword FLO is given, this transport is handled as a floppy disc.

  If the keyword BUL is given, this transport is handled as a bulk memory device.

  If the keyword LAR is given, then this transport is handled as a large-capacity moving head disc (333Ø-type).

[xcw]

- Parameter 1Ø is optional. It specifies the type of AFD, CLASSIC or Large Capacity disc drive that is being SYSGENed with this particular statement. The type is specified as a number in the range of Ø to 15 and the default value is 1. For a definition of this number, reference Appendix C-4.

EXAMPLES

```
DISCCHARACTER  M0,XZ,220,16,20,20,CYL,EXC
```

Specifies a standard disc (controller name is XZ) that requires a special handler, it has 220 bytes/sector, 16 sectors/track, 20 tracks/cylinder, 20 cylinders/transport, automatic cylinder advance, and exclusive use allowed.

```
DISCCH   M0,CD,128
```

Specifies a standard disc (controller name is CD that implies a cartridge disc) with 128 bytes/sector rather than the standard of 200.

```
DISCCH   H2,LD,256,64,19,411,CYL,EXC,LARGE,5
```

Specifies a standard disc (controller name is LD, that implies large-capacity moving-head disc) with 256 bytes/sector, 64 sectors/track, 19 tracks/cylinder, 411 cylinders/transport, automatic cylinder advance of overflow, exclusive use allowed of the transport, LARGE disc implying a 3330-type, and disc type of 5.

DEFINE THE TRANSPORTS FOR EACH CONTROLLER

The FMTRANSPORT statement is required and must be specified for each transport. The FMTRANSPORT statement can be used for either a File Manager transport or a non-File Manager transport. A physical transport, in general, is equivalent to a physical device to which one or more logical transports can be associated.

This statement is not required for those devices over which the File Manager cannot exert control (such as Process Input/Output devices, card readers, and teletypes) although a data structure known as the Transport Control Table (TCT) is generated automatically for these devices to maintain software compatibility when any SYSGEN macro of the form "...DEVICE" is found.

When the DISCCHARACTERISTICS statement is used, it must be given prior to the FMTRANSPORT statement for the same disc transport.

SYNTAX

```
                        1     2     3     4    5    6    7    8
     FMTRAN[SPORT]  tname tnum,cname      ,    ,    ,     [DPA]
                                                          [NOD]

                       9     1Ø      11
                     [bmc] [xcw] [PMO]
```

tname - Parameter 1 is the transport name (1- to 3-characters CAN-code). The name must be unique for all transports but there are no naming conventions (as there are for controllers).

tnum - Parameter 2 is the hardware transport number (Ø/1/2/3 for multi-transpor t controllers, Ø for single transport controllers) assigned to the transport as it is physically connected to the controller.

cname - Parameter 3 is the name of the controller to which this transport is connected (Refer to CONTROLLER statement).

- Parameters 4, 5, 6, and 7 are unused and should be defaulted.

[DPA]
[NOD] - Parameter 8 (optional) specifies the presence or absence of a dual-port adapter. The keyword DPA indicates that the controller shares control of the device with another (perhaps remote) controller and must take special precautions when requests access to

the device. The keyword NOD indicates that the controller has the device under dedicated control. If a Large Capacity Disc (333Ø-type) transport is to be configured without the pack change DCO, NOD must be used.

[bmc] - Parameter 9 (optional) is only used if the device is the Bulk Memory. The parameter is used to specify the number of Bulk Memory modules present. The default is 16 (or a fully expanded system).

[xcw] - Parameter 1Ø (optional) specifies the type of AFD, CLASSIC or Large Capacity disc drive that is being SYSGENed with this particular statement. The type is specified as a number in the range of Ø through 15 and the default value is 1. (For a definition of this number, refer to Appendix C-4.)

[PMO] - Parameter 11 is unused.

EXAMPLES

```
    FMTRANSPORT  MØ,Ø,CD
         Defines a disc transport associated with CONTROLLER CD.

    FMTRANSPORT  BØ,Ø,BM,,,,,DPA,8
         Defines a Bulk Memory that has a dual-port adapter
         attached, and contains 8 memory modules (1Ø24K words).
```

DEFINE EACH LOGICAL TRANSPORT ON A PHYSICAL TRANSPORT

The LOGTRANSPORT statement is used to define a logical transport on a physical transport. A structure known as a Logical Transport Table (LTT) is constructed for each LOGTRAN statement, therefore, more than one logical transport could be defined on a physical transport. A logical transport must be defined or contained on a single physical transport. There must be at least one LOGTRAN statement for every FMTRANSPORT statement. Partitions that are defined on DISCDEVICE statements should reference the logical transport name and not the physical transport name.

SYNTAX

```
       1         2     3     4    5     6    7
LOGTRA[NSPORT] lname pname strk ltrk [FMA] [vname] [rvd]
                           [FMP]
                           [NOF]

       8         9
     [EXC]     [PMO]
     [NOE]
```

lname - Parameter 1 is the CAN-code name of the logical transport. This must be a unique name within the system.

pname - Parameter 2 is the name of a physical transport defined by a FMTRANSPORT SYSGEN statement.

strk - Parameter 3 is the actual starting track address for this logical transport.

ltrk - Parameter 4 is the size in number of contiguous tracks of the logical transport being defined.

[FMA]
[FMP]
[NOF] - Parameter 5 is optional and is used to define the type of influence the File Manager (FM) exercises over this logical transport. If the keyword FMA is given (or the parameter is defaulted), then the FM always has complete and exclusive control of the transport. If the keyword FMP is given, then FM control of the logical transport is possible. Under either condition, the LTT is constructed so as to allow the FM to control the logical transport. Under FMP, OC Directives can be used to place/remove FM control of the logical transport. When the FM has control of the logical transport, BIOS access of the logical transport is precluded in order to protect FM-

formatted information of which BIOS has no knowledge. Further, the system will prevent the mounting of FM volumes on BIOS logical transports and conversely, again, with the objective of information protection. If the keyword NOF is given, a non-File Manager format LTT is generated.

[vname] - Parameter 6 is optional. It specifies the initially mounted logical volume name (which may be either a 1- to 6-character string or a 1- to 12-digit hexadecimal string preceded by the # sign). If this parameter is defaulted, no automatic logical volume mounting will be required at system start-up.

[rvd] - Parameter 7 is optional. It specifies the number of words to be reserved for a resident logical volume directory. When a logical volume is freshly mounted, as many entries as will fit are read into this space and maintained, as these particular files are updated. If this parameter is defaulted, a resident logical volume directory will not be present for this logical transport.

[EXC]
[NOE] - Parameter 8 is optional. It specifies the ability to take exclusive use of any partition or the entire logical transport. If the keyword EXC is given, taking exclusive use will be allowed. If the keyword NOE is given, taking exclusive use will not be permitted. The default is EXC.

[PMO] - Parameter 9 is optional. It specifies that this logical disc transport will have a logical volume permanently mounted on it. If the logical volume is not mounted at system start-up an appropriate message will be output and start-up will await mounting of the volume.

Rules for statement ordering:

1. The LOGTRANSPORT statement(s) must immediately follow the FMTRANSPORT of the disc to which they apply.

2. LOGTRANSPORT statements must be listed in the SYSGEN in the same order as their respective logical transports are defined on the physical disc.

EXAMPLES

```
FMTRANSPORT     PMØ,Ø,LX
LOGTRAN         MØ,PMØ,Ø,4ØØ,FMA
LOGTRAN         M1,PMØ,4ØØ,4ØØ,FMP
```

NOTES:

- o Overlapping of logical transports is not permitted and will result in an error message.

- o The existence of areas of a physical disc not contained in some logical transport will result in an error.

- o Logical transports must be defined contiguously and must cover the entire physical transport.

- o Logical transports are intended to be used on mixed-media drives (fixed and removable media).

DEFINE EACH DISC DEVICE IN SYSTEM'S LOGICAL DEVICE LIST

The DISCDEVICE statement is optional and, if present, specifies the name and size of a fixed, single-extent partition (contiguous number of tracks) on the referenced logical disc transport. These partitions are known as "logical devices" and are independent of the actual pack (volume) that is mounted on the transport. Once defined, the logical device may be associated with a logical file name and manipulated by a task that has made such an assignment. The entire logical transport cannot be manipulated in this fashion since it may not be assigned a logical device name but an equivalent capability can be arrived at by the subterfuge of defining a partition that includes all the tracks of the volume.

SYNTAX

```
                         1     2     3     4      5     6
    DISCDE[VICE]     dname tname ntk [geom] [bps] [atk]

                        7       8
                     [xutsk] [NOW]
                     [-infl] [WOR]
```

dname - Parameter 1 is the logical device name (1- to 3-characters CAN-code). The name must be unique for all devices (refer to DEVICE/COMDEVICE statements) in the system. Appendix C lists recommended naming conventions for these devices.

tname - Parameter 2 is the logical transport name on which this logical device is being defined and must already have been defined by an FMTRANSPORT statement.

ntk - Parameter 3 is the size in number of contiguous tracks of the partition being defined. If Parameter 6 is missing, the starting track of the partition will be the track immediately following the last track address of the partition defined by the last DISCDEVICE for this logical transport. The starting track for the first DISCDEVICE statement for a particular transport will be the value passed from the LOGTRANSPORT statement.

[geom] - Parameter 4 is optional. It specifies the addressing geometry for the logical device. It

describes a mapping of the logical sector address (implicit in the File Position Index in the User File Table associated with the REX service) into a physical sector address. A value of 1 for this parameter implies a 1:1 mapping which means that the logical sector address and the physical sector address on the track are equivalent. This is the value that must be used for partitions that hold Quick-Load modules. The default is SET to 4 using the equate name of GEODEF. If you wish to override the value of this default, SET GEODEF to the desired legal value prior to the DISCDEVICE statements.

Example:

```
GEODEF SET 20
DISCDE  AJC,L0,10
DISCDE  AMC,L0,SO
```

Causes both AJC and AMC to have a geometry of 20.

[bps] - Parameter 5 is optional. It specifies the bytes/sector value. If the given value is zero (0), then an "unlimited" block size is implied. In particular, the logical devices that hold Quick-Load modules have this parameter set to zero. The default for this parameter is defined by the DISCCHARACTERISTICS statement (that may itself have been implicitly invoked by the FMTRANSPORT statement).

[atk] - Parameter 6 is optional. It specifies the actual starting track address for this partition relative to the starting track of the logical transport. The presence of this parameter overrides the procedure outlined in Parameter 3 (ntk) for the automatic generation of the starting track address. This parameter would normally be present only when it is desirable that two or more partitions share some or all of their respective space on the transport.

[xutsk]
[-infl] - Parameter 7 is optional. It specifies either a task name or a negative influence limit. If a task name is given, the designated task will be given permanent exclusive use of the device to the complete exclusion of all other tasks. If an influence limit number in the range of -000 to -255 is given, then only tasks that have an influence limit number that is less than or

equal to this parameter will be permitted to use this logical device. The default is an influence limit number of -255.

[WOR]
[NOW] - Parameter 8 is optional. It specifies whether an End-of-Record (EOR) mark is to be written on the last sector for every write operation performed on this transport. If the keyword WOR is given, then an EOR will be written. If the keyword NOW is given, then an EOR will not be written. Once EOR marks have been written, then a read operation will "not" input a larger block than that which was written. The EOR marks are generally used only when Parameter 5 specifies an unlimited (and therefore variable) record size and the routine reading wishes to use the hardware EOR detection capability to delimit the record size. Default is NOW.

Use of the DISCDEVICE statement generates an assembly symbol of STKxxx where xxx is the DISCDEVICE dname. This symbol can be used as the "atk" parameter when defining overlapping partitions in that STKxxx is SET to the starting track number for partition xxx.

EXAMPLES

```
DISCDEVICE  AJC,M0,10
```

Define logical device to hold job procedures for all batch tasks.

```
DISCDE   BJW,M0,50,2,,,B
```

Define logical device for use only by the B task which is used for expansion of the job procedures.

```
DISCDE   QCK,M1,100,1,0
```

Define logical device to hold quick-load modules (that is, geometry = 1 and record size = unlimited).

```
DISCDEVICE  MZ,M2,25,,,,-125
```

Define logical device for use only by tasks that have an influence limit number in the range of 0 to 125.

DEFINE A TRANSIENT LOGICAL DEVICE POOL

The TLPDEFINE statement is optional. It specifies the logical file name to be used to bring in transient partition definitions and the size of the free pool of blank partition data structures. Notice that Parameter 1 is a logical FILE name and not a logical device name. The logical file name needs to be provided by a LOGFILE statement in PART 3 of the SYSGEN assembly as part of the definition of the system Exceptional Condition (X) task.

The free pool of blank partition data structures is maintained as a linked list so that as transient partition definitions are brought into memory or File Manager Partitioned Data Files are opened or temporary FAT/LDT pairs are defined, an entry from the list is stripped off, filled out, and linked into the logical device list. If this statement is not supplied, a default configuration of lfname=NO and number=1 is generated.

SYNTAX

```
                        1         2
     TLPDEF[INE]     [lfname] [number]
```

[lfname] - Parameter 1 is optional. It specifies the logical file name to be used when bringing transient partition definitions into memory. This file name must be associated with some logical device by a LOGFILE statement in PART 3 of the SYSGEN assembly as part of the global files within the Exceptional Condition (X) task. The default logical file name is TPD.

[number] - Parameter 2 is optional. It specifies the number of blank partition data structures that are present in the free pool. These blank structures are linked together with the head of the list pointed to by UCBTLP in the Upper Control Block. This number needs to be of sufficient size to accommodate all the transient partition definitions, all open File Manager Partitioned Data Files, and all temporary FAT/LDT pairs that are concurrently present in memory. The default value for the number of entries is 60.

EXAMPLES (only one usage permitted per assembly):

```
TLPDEFINE
     Use defaults (logical file name = TPD and free pool size
     = 6Ø).

TLPDEF   XYZ,75
     Use  logical  file name XYZ and set up a free pool of 75
     blank structures.

TLPDEFINE  NRD
     Use  logical file name NRD and default free pool size to
     6Ø entries.
```

DEFINE EACH NON-DISC DEVICE IN THE SYSTEM LOGICAL DEVICE LIST

The DEVICE statement must be given for each non-disc logical device to be defined. In most cases, there will be only one logical device define per controller while in other cases, several logical devices will be defined for a single controller (consider, for example, magnetic tape -- which can have up to four tape drives for one tape controller). This statement will generate a data structure known as the Logical Device Table (LDT). Where necessary, this statement will also implicitly generate the Transport Control Table (TCT) to maintain compatibility with the disc transport data structures. The implicitly generated TCT will be a much abbreviated version of that used with disc transports. Certain logical devices may be configured to operate under the control of the File Manager (FM). In this case the TCT for the device must be generated by an explicit, user-provided invocation of the FMTRANSPORT statement. When this is done, the second parameter of this statement "must" specify the transport name as given on the FMTRANSPORT statement. Otherwise, the second parameter for this statement is the controller name as given on the CONTROLLER statement.

SYNTAX

```
                         1      2       3     4      5      6      7
     DEVICE            dname {cname} [bpr] [opt] [xutsk] [bof] [null]
                             {tname}             [-infl]
```

dname - Parameter 1 is the logical device name (1- to 3- CAN-code characters) of the form "ddx", where dd equals device type and x equals a digit to make the name unique for a given type of device as well as for all the logical devices in the system (see DISCDEVICE and COMDEVICE statements). The rationale for using a standard logical name is to allow the system to provide automatic default values for some of the parameters on this statement.

{cname}
{tname} - Parameter 2 is either the name of the controller to which this logical device is connected (see CONTROLLER statement) or the name of the transport on which this logical device is being defined (see FMTRANSPORT statement).

[bpr] - Parameter 3 is optional. It specifies the natural block size (bytes/record) for the device. If the given value is zero (Ø), then an "unlimited" block size is implied. The default for this parameter is based on the device type, which is in turn based on the device name, (refer to Parameter 1). Appendix C lists standard logical device names.

[opt] - Parameter 4 is optional. It specifies modifications to the contents of the logical device characteristics word in the data structure (LDT) for this device. It specifies certain options that may be examined by the device handler when the handler supports similar but not identical devices. Again, the default for this parameter is based on the device type which is based on the device name and Appendix C lists the standard values for these options.

[xutsk]
[-infl] - Parameter 5 is optional. It specifies either a task name or a negative influence limit. If a task name is given, the designated task will be given permanent exclusive use of the device to the complete exclusion of all other tasks.

If a negative influence limit number in the range of -ØØØ to -255 is given, then only tasks that have an influence limit number with absolute value less than or equal to the absolute value of this parameter is permitted to use this logical device. The default is a negative influence limit number of -255.

[bof] - Parameter 6 is optional. It specifies the bottom of form simulation support count. This is the number of lines per page. Default is to use hardware bottom of form signal.

[null] - Parameter 7 is optional. It specifies the trailing null count if the number of trailing nulls required by this device exceeds those configurable by the device characteristics options. (This parameter applies only to those devices under control of the teletype handler.)

EXAMPLES

DEVICE TY,TT,82
Define logical device for extra wide carriage teletype keyboard/printer.

DEVICE PR,TT,
Define logical device for paper-tape reader using teletype controller.

DEVICE TP,TT
Define logical device for paper-tape punch using teletype controller.

DEVICE TY1,TT1
Define logical device for another teletype keyboard/printer on another controller.

DEVICE MT1,MT,,,B
Define logical device for magtape drive 1 under permanent exclusive use of the batch task B.

DEVICE MT2,T1,,,-100
Define logical device for magtape drive 2 on transport T1 for use by tasks that have an influence limit number in the range of 0 to 100.

DEVICE

DEFINE EACH "IMAGINARY" DEVICE OF EACH SYMBIONT CONTROLLER

This form of the DEVICE statement is optional and, if present, specifies the name of an "imaginary" logical device and the name of the symbiont task that processes transfer requests to the logical device. Such a device may be given the characteristics (for example, name, record size, options) of a real peripheral device so that the using task (program) is fooled into thinking that it is communicating with a real device. Following a request to the imaginary logical device, it becomes the job of the symbiont task to process the request and supply the appropriate simulation of a real device within the framework of BIOS.

SYNTAX

```
                        1     2     3     4      5       6      7
     DEVICE           dname task [bpr] [opt] [xutsk] [bof] [null]
                                               [-infl]
```

dname - Parameter 1 is the logical device name (1- to 3- CAN-code characters). The name must be unique for all devices in the system (refer to the DISCDEVICE/COMDEVICE statements). Appendix C lists recommended naming conventions for the standard imaginary devices.

task - Parameter 2 is the name of the symbiont task to which this device is attached (refer to the SYMCONTROLLER statement).

```
*****************************************************************
*  Parameters 3 through 7 are the same as those defined         *
*  for the DEVICE statement.   Please reference that            *
*  description for further information.                         *
*****************************************************************
```

EXAMPLES

```
DEVICE  TY2,S,80
```

Define logical device for an imaginary CRT terminal with an 80-character carriage.

```
DEVICE  LPX,S,134
```

Define logical device for an imaginary line printer with 132-column carriage.

```
DEVICE  TYA,T,74,#1800
```

Define logical device for an imaginary teletype with special characteristic.

DEFINE EACH COMMUNICATIONS "DEVICE" IN LOGICAL DEVICE LIST

The COMDEVICE statement is optional and is only used in the generation of a system that includes a(any) COMCONTROLLER statement(s). One or more of these statements may be associated with a COMCONTROLLER statement. This statement internally invokes the DEVICE statement to generate the appropriate data structure (LDT) and uses Parameters 6, 7, ..., 14 of the COMDEVICE statement to generate an LDT containing the necessary information.

SYNTAX

```
                     1      2      3     4      5        6
COMDEV[ICE]     dname cname [bpr] [opt] [xutsk] [wdog]
                                        [-infl]

                  7       8        9      10       11
                [csiz] [parity] [stop] [echo] [nulcnt]

                 12      13     14    15
                [baud] [sync] [bof] [plc]
```

dname - Parameter 1 is the logical device name (1- to 3- CAN-code characters) of the form "dxx", where it is recommended, d equals A for asynchronous devices, S for synchronous devices or M for MAXNET devices, and xx equals any characters necessary to build a logical device name that is unique (refer to DISCDEVICE/DEVICE statements).

cname - Parameter 2 is the name of the communications controller to which this logical device is connected (see COMCONTROLLER statement).

[bpr] - Parameter 3 is optional. It specifies the natural block size (bytes-per-record) for the device. If the given value is zero, then an "unlimited" block size is implied. The default for this parameter is based on the device type, which is in turn based on the device name (refer to Parameter 1). Appendix C lists standard logical device names.

The block size is normally used to specify the carriage width of an asynchronous terminal device. Printing terminals are normally given a block size that is two (2) characters longer than their carriage width so that carriage control characters can be transmitted as part of the block. For CRT display terminals, the block size is normally specified the same as the carriage width. With most CRT's, an automatic carriage return/line feed will occur when the end of line is reached, and if the user outputs a line feed, a double line feed will result causing double spacing.

The block or unit record size for a teletypewriter is normally specified as 74, and the block size for CRT's is normally specified as 8Ø.

[opt] - Parameter 4 is optional. It specifies modifications to the contents of the logical device characteristics word in the data structure (LDT) for this device. It specifies certain options that may be examined by the device handler when the handler supports similar but not identical devices. Again, the default for this parameter is based on the device type, which is based on the device name. For MAXNET options see the MAXNET reference manuals. For plc option see Parameter 15 below.

[xutsk]
[-infl] - Parameter 5 is optional. It specifies either a task or a negative influence limit. If a task name is given, the designated task will be given permanent exclusive use of the device to the complete exclusion of all other tasks. If an influence limit number in the range of -ØØØ to -255 is given, then only tasks which have an influence limit number that is less than or equal to this parameter will be permitted to use this logical device. The default is an influence limit number of -255.

[wdog] - Parameter 6 is optional. It specifies the maximum time in basic clock interrupt periods that an operation to this device will be allowed to stay at the head of the controller queue. This parameter has no meaning if Parameter 4 of the COMCONTROLLER statement specifies UNTIMED. The default value is Ø. Although this value is declared in this statement to maintain a compatible calling sequence with prior revision, the value of this parameter is moved to the associated PDT for this device as declared in the COMCONTROLLER statement.

[csiz] - Parameter 7 is optional. It specifies the data character (frame) size in bits. The range of this parameter is 5, 6, 7, or 8 and represents the actual number of data bits transferred to/from memory for an input/output operation. The default is 8. For SDLC/HDLC this parameter must be specified as "DLC", and parameters 8, 9, and 10 take on new meanings as specified below.

[parity] - Parameter 8 is optional. It specifies the type of parity that will be generated/checked by the controller hardware or SDLC/HDLC receiver length. If the keyword EVEN is given, then even parity will be generated on output and even parity will be checked on input. If the keyword ODD is given, then odd parity will be generated on output and odd parity will be checked on input. If the keyword NONE is given, then no parity generation or checking will take place. The default is NONE.

If parameter 7 was specified as "DLC", this parameter specifies the receiver length (number of bits per character; that is, 1 through 8). There is a no default value for this parameter when it specifies the receiver length.

ASCII terminals that have the parity bit as always a space (0) should normally specify the character size as 8 and the parity as none. ASCII terminals that have the parity bit as always a mark (1) should change the parity generation/check option in the terminal to SPACE, EVEN, or ODD parity since marking parity is not supported by the hardware. Parity generation and checking may be performed in user coded software by specifying the character size one bit larger to accommodate the parity bit.

[stop] - Parameter 9 is optional. It specifies the number of stop bits or SDLC/HDLC transmitter length. The value of the parameter may be either 1 or 2. This parameter only has meaning for asynchronous or SDLC/HDLC devices. The default is 2.

If parameter 7 was specified as "DLC", this parameter specifies the transmitter length (number of bits per character; that is, 1 through 8). The default value for this parameter is 2 bits.

[echo] - Parameter 10 is optional. It specifies the presence and type of echo (character retransmission on input) desired for this logical device or SDLC/HDLC secondary address.

If the keyword HARD is given, then the hardware in the communications controller will be used to perform the echo function. (NOTE: Full duplex I/O can cause abort of output due to the inability of the hardware and software to simultaneously echo and output data.)

If the keyword SOFT is given, then the handler (for the 1905 communications controller only) will perform the echo function. If the keyword NOECHO is given, then no hardware or software echo will be supplied. This parameter only has meaning for full duplex asynchronous devices. The default is NOECHO.

If parameter 7 was specified as "DLC", this parameter specifies the secondary address in the range of 0 to 255. If not specified, secondary station zero will be assumed.

[nulcnt] - Parameter 11 is optional. It specifies the number of NUL characters (default=0) to be output prior to each message. These null characters allow non-hand-shaking devices the time necessary to complete previous operations before beginning a new one. This parameter is provided for asynchronous devices that require terminal dependent delays, an example being a carriage movement delay required for a printing terminal.

A MODCOMP Model 4205 terminal, a GE Terminet 300 terminal or a TI Silent 700 terminal requires 8 null characters to be output to provide for carriage control operations. The MODCOMP Model 4206 terminal or the GE Terminet 1200 requires that 32 null characters be output to provide for carriage control operations.

[baud] - Parameter 12 is optional. It specifies for the 4804, 4806, 4807, 4808, 1907, 1907A, 1908, and multifunction communications controllers only, the value of the line speed. This parameter may take on one of the following baud rates: 75, 110, 134 (indicating a true baud rate of 134.5), 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, or 19.2 (indicating 19.2K baud). If the keyword EXTD is given, then the timing signal is assumed to be externally supplied. If this parameter is unspecified, the hardware default

rate will be used. The default rate is optional but is normally set at 12ØØ baud. Note that the maximum selectable baud rate for the multifunction communications controller is 96ØØ.

[sync] - Parameter 13 is optional. It specifies the synchronous character for synchronous channels on the 19Ø7 controller only. The sync character is specified as a hexadecimal character pair (#16 for ASCII or #32 for EBCDIC). The default is #ØØ.

[bof] - Parameter 14 is optional. It identifies the bottom-of-form simulation support count. This is the number-of-lines per page. Default is to use hardware bottom-of-form signal. This function is designed for use on output (printer) devices or on output channels only.

[plc] - Parameter 15 is used only when the PLC option is selected (Bit 7 set in Parameter 4 above). When the PLC bit is set, parameter 15 identifies the Programmable Logic Controller type that is interfaced to this COMDEVICE.

For PLC type:	Set Bit:
Square D	11
Texas Instrument PM 55Ø	12
General Electric Series 6	13
Gould, Modicon	14
Allen-Bradley	15

EXAMPLES

```
COMDEVICE  AØØ,AØØ,8Ø,,,,8,NONE,1
     Full-duplex asynchronous.

COMDEVICE  AØ1,AØ1,8Ø,,,,8,NONE,1
     CRT terminal.

COMDEV  AØ2,AØ2,74,,,,7,EVEN,2
     Half duplex teletypewriter.

COMDEV  SB8,SB8,,,,1ØØ,8,NONE,,,,#32
COMDEV  SB9,SB9,,,,1ØØ,8,NONE,,,,#32
     Full-duplex 19Ø7 Bisync.
```

BRKCHA

DEFINE SYSTEM BREAK CHARACTER

The BRKCHARACTER statement is optional. It identifies the system break character used on the main console and remote terminals.

Break character processing is described in the MAX IV COMMUNICATIONS HANDLERS, System Guide Manual listed in the Preface. The system break character defaults to CONTROL "A" if this statement is not included in the SYSGEN program.

SYNTAX

```
                  1
BRKCHA[RACTER]  [character]
```

[character] - Parameter 1 is the System Break Character. It is optional and if not specified defaults to CONTROL "A". The system break character should not conflict with listing control characters or buffered device X/ON - X/OFF characters. Refer to the MAX IV COMMUNICATION HANDLERS, System Guide Manual listed in the Preface for further details.

EXAMPLES

```
BRKCHARACTER  #07
     Specify CONTROL "G" as the system break character.

BRKCHA
     Default the system break character to CONTROL "A".
```

GENERATES A LDT

The SIODEVICE statement must be specified for each DCS channel to be defined. This statement generates a Logical Device Table (LDT). The LDTs that are generated are resident in MAP 0.

SYNTAX

```
    SIODEV[ICE]  dname,cname[,bpr][,opt][,xtask]
                                        [,-infl]
```

dname - Parameter 1 is the logical device name (1 to 3 CAN code characters) and must be unique among all device names.

cname - Parameter 2 is the MUX controller name which was specified as parameter 1 of the associated MUXCONTROLLER statement.

[bpr] - Parameter 3 is an optional parameter which specifies the record size in bytes (bytes per record) for this device. Valid values are 0 through 510. If 0 (the default) is specified, the record size is considered unlimited. See also parameter 7 (MAXTC - maximum transfer count) of the SIOCONTROLLER statement.

[opt] - Parameter 4 permits modification of the contents of the logical device characteristics word in the data structure (LDT) for this device. It specifies certain options that may be examined by the specific handler when a handler supports many similar but not identical types of devices. This LDT option word is shown in the specific DCS handler chapters.

[xtask]
[-influ] - Parameter 5 is optional. It specifies either a task name or a negative influence limit. If a task name is given, the designated task is given permanent exclusive use of the device to the complete exclusion of all other tasks.

If a negative influence limit number in the range of -000 or -255 is given, then only tasks that have an influence limit number with absolute value less than or equal to the absolute value of this parameter are permitted to use this logical device. The default is a negative influence limit number of -255.

DEFINE DYNAMIC INTERVAL TIMER PACKET POOL SIZE

The DITPOOL statement is optional. It specifies the number of system dynamic interval timers. Dynamic Interval Timers are used by the system for features such as Automatic Terminal Listing Resume capability, Asynchronous Floppy Disc Protocol Synchronization, Ring Detect Line Disconnection delay, and Break Key processing. If this statement is not included in the SYSGEN program, one Dynamic Interval Timer will be included in the system for each of the following conditions:

- Non-MAXNET/Non-AFD COMCONTROLLER statements.
- BREAK supported LINEMONITOR statements.
- Ring Disconnect Delay supported LINEMONITOR statements.

SYNTAX

```
                       1
     DITPOO[L]      [number]
```

[number] - Parameter 1 is optional. It specifies the number of System Dynamic Interval Timers to be included in the system. The default is one for each of the conditions listed above.

EXAMPLES

```
     DITPOOL  5
          Define five Dynamic Interval Timers.

     DITPOO
          Default the number of Dynamic Interval Timers to the
          number determined by the system configuration.
```

LISTCO

DEFINE TERMINAL LISTING CONTROL INFORMATION

The LISTCONTROL statement is optional. It defines the system listing control characters and automatic listing resume time interval. Terminal Listing Control processing is described in the MAX IV COMMUNICATIONS HANDLERS, System Guide Manual listed in the Preface. If this statement is not included in the SYSGEN program, the system listing stop character defaults to CONTROL "S" and the system listing resume character defaults to CONTROL "Q".

SYNTAX

```
                          1     2     3
     LISTCO[NTROL]      [lcs] [lrc] [lri]
```

[lsc] - Parameter 1 is optional. It specifies the Listing Stop Character. The default for this parameter is #13 or CONTROL "S".

[lrc] - Parameter 2 is optional. It specifies the Listing Resume Character. The default for this parameter is #11 or CONTROL "Q".

[lri] - Parameter 3 is optional. It specifies the Automatic Listing Resume Time Interval. This parameter specifies the number of ticks in the interval to be used by the handler in automatically resuming a held output listing. The timer operates at the low clock level and begins whenever a stop-listing condition is detected upon starting up an output operation. The default for this parameter is 0, which indicates that listings are not to be automatically resumed.

The following characters are RESERVED and may not be used as Listing Stop/Resume Characters:

System Break Character - the default is "CONTROL A" (#01)

DELETE/RUBOUT (#7F)

Carriage Return (#0D)

Backspace (#08)

NUL (#00)

EXAMPLES

```
LISTCONTROL  #06,#07
```

Specify CONTROL "F" as the Listing Stop Character and CONTROL "G" as Listing Resume Character.

```
LISTCO   ,,500
```

Specify Automatic Listing Resume Interval as 500 ticks using the Default Listing Control Characters.

SELECT ALTERNATE CARD-CODE CONVERSION TABLE

The CARDCODE statement is optional. It indicates a desire to include an alternate card conversion code table for the Card Reader handler. The handler expects to read cards punched in ANSI Standard X3.26-197Ø Hollerith card code; therefore, the card conversion code table for this translation is always included. The ANSI Hollerith card code is the same as the code punched on an Ø29 keypunch except for four characters.

ANSI SYMBOL	Ø29 KEYPUNCH SYMBOL	INTERNAL ASCII
!	|	#21
[	¢	#5B
]	!	#5D
—	—	#5E

The system provides a standard alternate card conversion table for handling cards punched on an Ø26 keypunch. This statement will also accept a user-defined card conversion code table.

SYNTAX

```
                          1      2
     CARDCO[DE]        [code] [fname]
```

[code] - Parameter 1 is optional. It specifies the characters to be concatenated to the end of "CR" to generate an insert file name. From one to four characters may be used. The default is "Ø26".

[fname] - Parameter 2 is optional. It specifies the logical file name on which the insert file named in Parameter 1 resides. The value of the parameter may be from 1- to 3- CAN-codable characters. The default is MC.

EXAMPLES

```
     CARDCODE
          Select Ø26 as alternate.

     CARDCO   FLDA,MMC
          Select CRFLDA as insert file name and MMC as logical
          file name.
```

NOTE: Use of the alternate card conversion code table is selected at task run-time by setting either Task System Option Bit 6 or the DEVICE DEPENDENT OPTION (DDO) bit in the option word of the User File Table (UFT) associated with a REX call sent to the card reader.

LINEMON

DEFINE COMMUNICATION LINE MONITORING CHARACTERISTICS

The LINEMONITOR statement is optional. It is used only if a Break/Ring processing task is to be included in the system. This statement is not required for the Operator Communications Task and its terminal to detect Break. This statement is required for each communications line for which Break/Ring will be queued to some line monitoring support task. One such task is supplied by MODCOMP to provide for Remote Operator Communications and Auto Answer support (this task is called the Terminal Monitor Program - TMP); however, any number of user tasks may also exist in the system.

SYNTAX

```
                       1       2         3         4       5        6
     LINEMON        output [input] [options] [task] [prefix] [delay]
```

output - Parameter 1 is the logical device name on which BREAK, CONTROL "A", RING, and LINE DISCONNECT are to be detected. If the device is a full-duplex device, "output is the name of the device on the output channel.

[input] - Parameter 2 is only required if the device is a full-duplex device. If the device is half-duplex, either this parameter must not be entered or must be the same as Parameter 1. If the device on which BREAK, CONTROL "A", RING, and LINE DISCONNECT is to be detected is a full-duplex device, "input" is the logical device name of the device connected to the input channel.

[options] - Parameter 3 is optional. It specifies line monitoring characteristics as follows:

Break Support	BREAK	(#2000)
Ring Support	RING	(#1000)
Auto Task Line Connect Support	CONN	(#0800)
Auto Task Line Disconnect Support	DISC	(#0400)
Line Disconnect Delay Support	DELAY	(#0200)

Auto Task Line Connect Support, Auto Task Line Disconnect Support and Line Disconnect Delay Support require that Ring Support be optioned. The default option is BREAK support.

[task] - Parameter 4 is optional. It specifies the task name that is to be notified (by a packet queued to it) when a BREAK, RING or LINE DISCONNECT is detected from the supported device. All these task names must be unique and only 3 characters long. The default task name is TMP (Terminal Monitor Program).

[prefix] - Parameter 5 is optional. It specifies a 3-character CAN-code task name prefix to be used as the prefix to the initial attentive task name. The 3-character suffix to the task name will be the line number in decimal ASCII. A line number is based on the LINEMONITOR statement ordering. The line number of the first LINEMONITOR statement is "ØØ1". The line number of each additional statement is "ØØ2", "ØØ3", et cetera. If this parameter is not supplied then there will be no initial attentive task name.

[delay] - Parameter 6 is optional. It specifies the number of ticks to delay after detecting a line disconnection. The time delay operates at low clock level. This parameter is used in conjunction with Line Disconnect Delay Support (DELAY) only and the default delay in 3 seconds or 6ØØ ticks.

EXAMPLES

```
LINEMONITOR  AØØ,AØ1,BREAK,TMP
     Full-duplex device with break support.

LINEMO   AØ2,,BREAK,TMP
     Half-duplex device.

LINEMO   AØ4,AØ5,RING,TMP,BAT
     Full-duplex device with ring support, attentive task
     name is BATØØ1.

LINEMO   AØ6,,BREAK+RING+CONN+DISC+DELAY,TMP,BAT,6ØØ
     All supported options, disconnect delay of 6ØØ ticks.
```

ENDIOS

DECLARE END OF SECOND SYSTEM-GENERATION ASSEMBLY

The ENDIOS statement is required. It finalizes the basic input/output data structures and terminates the Macro Assembly (outputs an END statement) for this second section of the system generation source file. The resulting "incomplete" main program defines one-third (or one-fourth) of the standard system data structures and must be followed by the other Macro Assemblies which are defined in PART 3 (and optionally, PART 4).

SYNTAX

```
                          1            2
     ENDIOS          [MES[SAGES]] [STA[LLS]]
```

[MES] - Parameter 1 is optional. It specifies that an operator message is to be output whenever an operation is queued to a logical device whose controller is busy (that is, in the HOLD state).

[STA] - Parameter 2 is optional. It specifies that BIOS should not attempt to queue any further operations to a device whose controller is in the HOLD state. If the keyword STA is given, then a calling task that encounters this condition will be forced to relinquish CPU time until such time as the HOLD condition is cleared. The default is no relinquishing of CPU time of a HOLD condition.

EXAMPLES (only one usage permitted per assembly)

```
     ENDIOS  MES,STA
          Specify both options.

     ENDIOS  MESSAGES
          Specify just messages.

     ENDIOS
          Specify neither option.
```

2.3 PART 3 - TASK STRUCTURE BLOCK

The statements described in PART 3 must immediately follow the statements described in PART 2. A separate Macro Assembly step is used so that memory requirements to effect a system generation are reduced to an acceptable level. This assembly phase constructs several permanent data structures in the system nucleus that are used by the TASKMASTER and the system COLD START/RESTART routines as follows:

- A threaded list of all permanently resident Task Control Blocks (TCB) for permanent privileged tasks that share MAP Ø addressing space with MAX IV;
- A threaded list of TCB's for all prescheduled non-resident tasks, the TCB'S of which are added to the CPUQ at system cold-start/restart;
- A last-in/first-out (LIFO) stack containing the descriptions of all vacant addressing space in MAP Ø that can be used to build TCB's of transient non-resident tasks as well as other transient data structures.

LISTING TITLE - START OF NEW ASSEMBLY

This standard Macro Assembler statement is optional and can contain any message. Its purpose is to ensure that the SYSGEN listing starts on a whole page boundary -- and that the title specified appears on each page of the listing.

SYNTAX

```
                         1
     TTL              heading

heading             - Parameter 1 indicates the heading that appears
                      at the top of each page of this part of the
                      SYSGEN listing.
```

EXAMPLE

```
     TTL   MAX IV: TSKBLK
```

INSERT MACROS TO DEFINE THE MACRO LANGUAGE

Before the macro language that defines MAX IV task structures can be used, the prototypes that define the language must be inserted from an external file. This standard Macro Assembler statement performs this function and MUST be present.

SYNTAX

```
                   1      2
     INS          fname IVTASK
```

fname - Parameter 1 specifies the logical file name where the macro prototypes are cataloged. The MAX IV standard for this file is MC, the Common Macro Library. This file name must be assigned to a real file or device before it can be used.

IVTASK - Parameter 2 is the name of the particular macro-group used to define this macro language.

EXAMPLE (only one usage permitted per assembly)

```
     INS  MC,IVTASK
          Define macro language.
```

All the SYSGEN statements in PART 3 are defined by the MAX IV Task Structure Prototype. These definitions are included in the assembly by using this statement.

PGM

NAME OF PROGRAM

This standard Macro Assembler statement is optional and assigns a name to the program being assembled. This statement must follow the INS statement in order to override the standard default provided by the macro file inserted for this block of the SYSGEN.

SYNTAX

```
                        1
     PGM              name
```

name - A valid symbolic name as defined in the MODCOMP Assemblers, Language Reference Manual. GT.IV is the default name for the Task Structure Block.

SPECIFY EACH PRE-SCHEDULED NON-RESIDENT TASK

The PRESCHEDULE statement allows a permanent prologue TCB to be included in the CPUQ for each specified non-resident task. This allows for more rapid activation of non-resident tasks than would be possible if the TCB were constructed and queued dynamically (requires search of disc-resident directories of cataloged tasks). The task body and final TCB will still be loaded from disc, however. The data structure built by repeated use of this statement will instruct the system's COLD START/RESTART routines to schedule the indicated non-resident tasks in the manner indicated. The TCB's which result are thus not contained in the resident nucleus but are added to it at system start.

Each prescheduled task will have a collapsed TCB data structure generated at system start thus occupying about two pages of MAP Ø, however, each PRESCHEDULE statement will increment (by one) the number of TRANSIENT tasks defined by the TRANSIENTS statement.

The PRESCHEDULE statement must not be used in Systems using the sequential task loader. If MLSEQUENTIAL and PRESCHEDULE are both specified, the results will be unpredictable.

SYNTAX

```
                         1       2        3        4
     PRESCH[EDULE]  name [level] [file] [PER[M]    ]
                                          [NOP[ERM]]
```

name - Parameter 1 is the desired task name (1- to 6-CAN-codable characters).

[level] - Parameter 2 is optional. It specifies the desired priority level (in the range of Ø-255) of the task to be pre-queued--when the system is started. The default is 255.

[file] - Parameter 3 is optional. It identifies the logical file name from which the non-resident task will be loaded when activated or established. The default file name is LM.

[PER[M]]
[NOP[ERM]] - Parameter 4 is optional. It allows the prescheduled task to leave the system after first executing. Otherwise, its TCB becomes a permanent part of the CPUQ. The default is PER.

EXAMPLES

```
     PRESCHEDULE  B,,SM
          Define  the  B task with priority 255, from logical file
          SM, and permanent TCB.

     PRESCH   RDS,25,SM
          Define  the RDS task with priority 25, from logical file
          SM and permanent TCB.

     PRESCH   ABC,200,BM
          Define the ABC task with priority 200, from logical file
          BM and permanent TCB.
```

SPECIFY THE MAXIMUM NUMBER OF TRANSIENT NON-RESIDENT TASKS AND THE LARGEST CONTIGUOUS TCB SPACE

The TRANSIENT statement is optional. It specifies the desire on the part of the system user to allow for transient, non-resident tasks within this MAX IV system. Without this statement, only permanent, resident and pre-scheduled, non-resident tasks are supported.

SYNTAX

```
                          1         2
     TRANSI[ENT]     maxnumber addwords
```

maxnumber - Parameter 1 is required and specifies the maximum number of transient, non-resident tasks that can ever be concurrently active, established, or scheduled in the MAX IV system being generated.

addwords - Parameter 2 specifies the additional number of user words to be add ed to every TCB. Word TCBUSR of the TCB is initialized to Parameter 2. The user words start at word TCBUSR+1.

EXAMPLES (only one usage permitted per assembly)

```
     TRANSIENT   1Ø
```

Define the need for up to 1Ø transient tasks.

```
     TRANSI   5,4
```

Define the need for up to 5 transient tasks. Add 4 user words to every TCB.

IONODE

SPECIFY THE INITIAL NUMBER OF PRIVATE AND GLOBAL I/O NODES. ALSO SPECIFY THE MAXIMUM NUMBER OF GLOBAL I/O NODES.

The IONODE statement is required and specifies the initial I/O node allocation. All nodes are allocated from MAP Ø and placed in two linked lists - one for private usage and one for global usage. Private I/O nodes are allocated by TASKS when the task is loaded. Global I/O nodes are allocated when the task has no private nodes to use (all private nodes are in use). Private nodes are deallocated only when the task exits. Global nodes are deallocated by the handler/symbiont when an I/O operation is complete.

SYNTAX

```
                        1             2             3
     IONODE[S]       [initprivate] [initglobal] [maxglobal]
```

[initprivate] - Parameter 1 specifies the initial number of private I/O nodes to be allocated. Private I/O nodes for resident tasks are automatically added to this figure. When a task exits all pages with all I/O nodes deallocated in them are deallocated by the exiting task except those pages that contain the initial private nodes.

[initglobal] - Parameter 2 specifies the total number of global I/O nodes.

[maxglobal] - Parameter 3 specifies the maximum number of Global nodes that can ever be allocated.

EXAMPLES (only one usage permitted per assembly)

```
     IONODES   3Ø,1ØØ
```

30 initial private I/O nodes, 1ØØ total global I/O nodes.

```
     IONODES   3Ø,,1Ø
```

3Ø initial private I/O nodes, and any one task can have up to 1Ø global nodes.

SPECIFY THE NUMBER OF SYSTEM UFTS

This SYSUFT statement is optional. It specifies the number of system UFTs to be included in the MAX IV system.

SYNTAX

```
                            1
     SYSUFT               number
```

number - Parameter 1 specifies the number of system UFTs. This number should be equal to or greater than the largest number of concurrent REX MESSAGE operations.

EXAMPLES (only one usage permitted per assembly)

```
     SYSUFT   20
          Specify 20 system UFTs.
```

SPECIFY INITIAL ATTENTIVE TASK NAME FOR "OC" TASK

The ATTENTASK statement is optional. It specifies the name of the task that the Operator Communication (OC) task will consider requires service, or the task that requires attention following system COLD START. If this statement is missing then the name of the attentive task will be OC, itself.

SYNTAX

```
                       1
     ATTENT[ASK]    taskname
```

taskname - Parameter 1 is required and specifies the name of the task (1- to 6- CAN-codable characters) that is to be the initial attentive task.

EXAMPLES

```
     ATTENTASK   B
          Define the B task as the initial attentive task.

     ATTENT   OC
          Define the OC task as the initial attentive task.
```

SPECIFY USER MAP ZERO WORK QUEUE

The WORKQUEUE statement is optional. It specifies an additional MAP Ø resident work queue. The user may include in the system resident tasks or resident routines that can make use of one or more of these queues. The MAX IV Exceptional Conditions tasks, X and FMX, make use of queues of this type and have reserved the queue names XCT and FMX. Their queues, however, are automatically included in the system when the user configuration demands them. The module US.QUE contains all the necessary enqueue, dequeue, and postprocessing routines for complete queue manipulation.

SYNTAX

```
                     1    2     3      4
     WORKQU[EUE]    nam [eng] [deq] [lockout]
```

nam - Parameter 1 defines the work queue name and must be 1- to 3- CAN-code characters. The names XCT and FMX may not be used.

[enq] - Parameter 2 defines a 1- to 3- CAN-codable suffix to form the enqueue postprocessing routine name WE$enq. The default is WE$XCT.

[deq] - Parameter 3 defines a 1- to 3- CAN-codable suffix to form the dequeue postprocessing routine name WD$deq. The default is WD$XCT.

[lockout] - Parameter 4 defines the interrupt lockout level for queue manipula tion. The default is #F.

EXAMPLES

```
     WORDQUEUE   ABC
```

Specifies word queue ABC. Enqueue and dequeue routines are defaulted. Lockout level is defaulted.

```
     WORKQU   DEF,GHI,JKL
```

Specifies workqueue DEF, enqueue post-processor WE$GHI, dequeue post-processor WD$JKL and lockout level #F.

ROLLER

SPECIFY THE EXISTENCE OF THE ROLLER

The ROLLER statement is optional. It specifies that the ROLLER task is to be added to the system being generated. Without this statement, only as many transient and prescheduled, non-resident tasks as can actually fit into available actual memory can be active at any time. Activating tasks in excess of this number, which will vary according to the size of the respective tasks, will cause the task(s) being activated to be "stalled" pending the availability of memory space.

When the ROLLER task has been configured, any number of tasks may be activated at any one time and the ROLLER task sees that only those tasks of highest priority are allocated memory space and only those tasks of lowest priority are stalled. Further, the ROLLER task re-scans the allocation of memory each time a task exits to ensure that once again the tasks of highest priority can run and only those of lowest priority are stalled.

When the ROLLER statement is specified, and the MAP Ø resident ROLLER is optioned (when Parameter 2 is defaulted), the following TCB configuration is automatically generated. The configuration statements listed below are explained later in this chapter.

```
     TASK             ROL,255,SM
     PECULIAR         AUTOSTART
     STACKS           15Ø,12
     BUFFER           128
     SYSPAGES         1
     IOPERATIONS      2
     VARIABLES
     FINTASK          ROL
```

SYNTAX

```
                          1          2         3
     ROLLER           [priority] [MAP[PED]] file
```

[priority] - Parameter 1 is optional. It specifies that the normal technique of changing the ROLLER task's priority according to work required is to be overridden. If this parameter is supplied then the ROLLER task initially changes its priority, during ROLLER task startup, to this value and remains fixed at this priority. If this parameter is specified then a task variable named PRI will be added to the ROL task's resources. Its value will be equal to this parameter. In addition the ROL task's system option MAP will be set.

This parameter is ignored if the non-MAP Ø resident ROLLER is optioned (MAPPED specified for Parameter 2 - see below). Instead, the non-MAP Ø Resident ROLLER must be cataloged with the task variable PRI. The value of the task variable is the priority at which the non-MAP Ø resident ROLLER is to run.

[MAP[PED]] - Parameter 2 is optional. If MAP is specified, the non-MAP Ø resident ROLLER is to be configured in the system. If the MAP Ø resident ROLLER is to be configured this parameter must be defaulted.

file - Parameter 3 is the logical file name from which the non-MAP Ø resident ROLLER is to be activated. The default value is SM. This parameter is ignored if the MAP Ø resident ROLLER is optioned.

EXAMPLES

```
ROLLER
```

Request floating position MAP Ø resident ROLLER TASK presence.

```
ROLLER   1Ø1
```

Request fixed position MAP Ø resident ROLLER TASK presence.

```
ROLLER   ,MAPPED
```

Request non-MAP Ø resident ROLLER TASK. If it is cataloged with the task variable PRI it remains fixed in the CPUQ, otherwise, it floats the CPUQ.

NOTE: The number of tasks is ultimately limited by the file space on disc committed to the RO (ROLLER) file and the number of transient TCB's specified.

WARNING

The presence of the ROLLER and the Round-Robin TaskMaster does not make a time sharing system. For example, if three tasks at a given priority level are rolled out and a fourth task at the same priority level but higher in the CPU queue is in I/O hold waiting for input from a terminal, the I/O task does not get rolled out because it is higher in the CPUQ so that one of the rolled out tasks may get rolled in.

FMGRX

SPECIFY THE FILE MANAGER "X" TASK (FMX)

The FMGRX statement is optional. It specifies that the FILE MANAGER Exceptional Condition task (FMX) is to be included in the system. This statement must be specified if the FILE MANAGER is present.

When this statement is specified, the following TCB configuration is automatically generated. The configuration statements listed below are explained later in this chapter.

```
     TASK            FMX,255,Ø,SM
     PECULIAR        AUTOSTART,NODELAY,UNABORTABLE
     STACKS          145,2Ø
     BUFFER          Ø
     SYSPAGES        8
     FCBDEFINE       2
     LOGFILES        1
     IOPERATIONS     2
     FINTASK         FMX
```

SYNTAX

```
     FMGRX
```

EXAMPLE

```
     FMGRX
          Request the FILE MANAGER "X" Task.
```

2.4 SPECIFY EACH PERMANENTLY RESIDENT TASKS SYSTEM NUCLEUS (MAP Ø)

A series of statements follow which can uniquely define the characteristics of one or more permanently resident tasks. These tasks are always scheduled in the CPUQ, are privileged, and use MAP Ø (the system's) virtual space. Being privileged, however, they may select and use memory maps or map image tables of other tasks, and may address anywhere in actual memory directly. The TCB's and program bodies of these tasks are permanently embedded within the system nucleus and need not be on page boundaries.

The statements that define each task must be arranged sequentially. The first statement "TASK" begins the definition of a task and the last statement "FINTASK" finishes the definition of a task. Optional statements may appear between these two control statements when the default values of task characteristics are not desired. Each statement is defined below under separate headings.

The resident task X (called the Exceptional Condition task) is the only required resident task. All others may be non-residents. Often, the OC task (Operator Communications), RDS task (input spooler) and S task (output spooler) are resident too. The IDLE task, which is a partial task, is included in all systems automatically and its presence and characteristics need not be specified. The ROLLER task (ROL) is an unusual resident task whose TCB appears and then disappears at various priority positions in the CPUQ. It is configurable with the ROLLER statement.

The resident tasks must be specified in descending priority order (Ø to 255). Thus the X task must be specified first and the SYSGEN process automatically makes the IDLE task last.

TASK

SPECIFY EACH RESIDENT TASK'S NAME, PRIORITY, FILE

The TASK statement MUST be present to define at least the X and OC tasks and is used to begin the definition of a permanently resident task. The statements that follow it further define the specified task.

NOTE: All resident tasks must run in the pipeline mode on a CLASSIC.

SYNTAX

```
            1              2               3          4
    TASK tname       [priority]      [influence] [file]
```

tname - Parameter 1 is the name of the task (1- to 3- CAN-codable characters). It must be unique among all task names. Only non-resident tasks can have 6-character names.

[priority] - Parameter 2 is optional. It specifies the desired priority level in the range of 0-255 of the task when MAX IV is initially started or restarted. The priority of a resident task can be changed by the CHANGE service but it will revert to its initial priority whenever the system is cold-started. Default is 255.

[influence] - Parameter 3 is optional. It specifies the highest priority level at which this task is allowed to execute (refer to the CHANGE service description contained in the MAX IV Operator Communications, System Guide Manual and MAX IV Executive (REX) Services, System Guide Manual). The influence limit also determines to what extent this task can influence the execution of other tasks -- or be influenced by the execution of other tasks. Default is 255.

[file] - Parameter 4 is optional. It specifies the name of the file to be used for all overlays when a file name is not specified explicity on a load-overlay request. This parameter is rarely used with a resident task since resident tasks are advised not to overlay themselves unless their original transfer address code survives and to use only buffers which they have supplied. Default is LM.

EXAMPLES

```
TASK   X,ØØØ,ØØØ,SM
```

Define system Exceptional Condition (X) task with priority = Ø and influence = Ø and file name = SM.

```
TASK   OC,127,ØØØ,SM
```

Define system Operator Communication (OC) task with priority = 127 and influence = Ø and file name = SM.

SPECIFY CHARACTERISTICS FOR THE RESIDENT TASK

The PECULIAR statement permits certain "peculiar" characteristics to be given to the resident task currently being defined. These characteristics are usually reserved for system tasks but can be specified by the user for special applications. Any or all of the keywords may be specified as parameters of the statement (in any order). Non-resident tasks can have additional "peculiarities" (for example, PRIVILEGED, UNROLLABLE, and BATCH) that need not be specified for permanently resident tasks since they always execute in the privileged state and are unrollable.

If the user wishes the access time benefits of residency for an unprivileged task, then the PRESCHEDULED statement should be used to define it and the ESTABLISH service should be used to make an otherwise non-resident task become "apparently" resident. Such a task must additionally be cataloged with an UNROLLABLE peculiarity if it is to be truly resident since an established task under MAX IV is normally still rollable although only one copy of the program will be loaded.

SYNTAX

```
                                1                  2
     PECULI[AR]        [characteristic] [characteristic]

                                3                  4
                       [characteristic] [characteristic]
```

[characteristic] - Parameters 1, 2, 3 and 4 are optional. Each of the parameters is a keyword and may be specified in any order. The meaning of each keyword is explained below. There are no defaults.

[UNA[BORTABLE]] UNABORT is similar to the BATCH characteristic for non-resident tasks but is useful only for resident tasks. It causes a task that exits or aborts to immediately restart at its original root transfer address (coding at this point must survive any previous overlays). The standard Operator Communications (OC) task uses this feature, since depressing the CONSOLE INTERRUPT switch always aborts this task. The standard "X" task must also be given this peculiarity.

[REM[OTE]] REMOTE causes a task that calls the MESSAGE or WAIT services (with HOLD option), to not enter the HOLD state but instead, to accept a directive from a remote terminal assigned to its local OC or CO file. The received character string is then analyzed to see if it is one of a subset of operator directives (/RESUME or /ABORT). If equal, the RESUME or ABORT routine is then executed on behalf of that task.

[AUT[OSTART]] AUTOSTART makes the system COLD START routines activate this task after any system start/restart. This frees the user's custom restart routine (U$XINI) of this duty. The standard OC task may or may not have this characteristic. The standard X task does not need it since the system always starts this task first so that it can execute the routine U$XINI at its level to complete system start-up routines.

[SYM[BIONT]] SYMBIONT specifies that the task is a symbiont and must be linked to its resident symbiont controller table by the system's COLD START routine.

[RIO[COMPLETE]] RIOCOMPLETE specifies that anytime the task completes an I/O request it is to be resumed. If it is suspended and if it is active, it is to have a secondary resume posted to it.

EXAMPLES

```
PECULIAR   UNABORTABLE
     Define task as UNA[ABORT].

PECULI   REM,AUT
     Define task as REM[OTE], AUT[OSTART].

PECULIAR   SYM
     Define task as a SYM[BIONT].
```

BUFFER

SIZE EACH RESIDENT TASK'S SERVICE BUFFER

Every task that executes certain system services (MESSAGE, DUMP) must have a service buffer of at least 48 words. If the task executes the services LOVER, ACTIVATE, or ESTABLISH, it must have a service buffer of at least one disc sector (100 words or 128 words). The BUFFER statement specifies the size of this buffer. Non-resident tasks generally have their service buffers allocated/deallocated dynamically as they are needed. This feature can optionally be given to resident tasks by specifying a buffer size of zero (0) words with this statement. However, if the reason a task is made permanently resident is so that it can guarantee that its memory resources are always available to it, this practice is not recommended. The X task and OC task should be given the default permanent service buffer size.

SYNTAX

```
                     1
     BUFFER        [words]
```

[words] - Parameter 1 is optional. It specifies the number of words to allocate for the permanent service buffer. The default (and minimum non-zero) value is 256 words.

EXAMPLES

```
     BUFFER   256
          Define 256 word service buffer.

     BUFFER
          Define 256 word service buffer.

     BUFFER   0
          Define dynamic service buffer.
```

SIZE EACH RESIDENT TASK'S PUSH STACKS

The STACKS statement defines the size of the task's two LIFO push-stacks that it must have to call executive services. The default size is 128 words for the main stack and 12 words for the smaller PSD stack. (These SYSGEN default values are different from those in the Task Overlay Cataloger.) (mwords) is used for register saving and general reentrant storage needs of reentrant REX services and the PSD stack (pswords) facilitates nesting of REX services. Non-resident tasks may make their main stack dynamically extensible/retractable (up to 256 additional words) should the initially specified size be insufficient. This feature may also be used by resident tasks if the EXTENSIBLE keyword is specified. In this case, a main stack size (mwords) of zero (Ø) causes this stack to allocate a one page (256 word) stack in the virtual space pool permanently at system start/restart. If the reason for making a task permanently resident is so that it can guarantee that its memory resources always be available to it, the practice of giving resident tasks EXTENSIBLE stacks is not recommended. The X task and OC task should be given a permanent main stack with the default size of 128 words in normal systems.

SYNTAX

```
                            1          2            3
     STACKS             [mwords] [pswords] [EXT[ENSIBLE]]
                                           [NON[EXTENS ]]
```

[mwords] - Parameter 1 is optional. It specifies the size of the main LIFO push stack. The default value is 128.

[pswords] - Parameter 2 is optional. It specifies the size of the PDS LIFO push stack. The default value is 12.

[EXTENSIBLE]
[NONEXTEN] - Parameter 3 is optional. It specifies whether the main LIFO push stack is extensible when the initially specified allocation (that is, that in Parameter 1, above) is filled.

WARNING

STACKS are not mapped into MAP Ø except while the task is currently running or has a resident TCB extension (permanent MAP Ø task). Tasks may not directly modify another task's stack UFTs, and further, I/O buffers cannot be placed in the push stack.

EXAMPLES

```
STACKS  256,32,NON
```

Define 128 word main stack, 32 word PSD stack, main stack non-extensible.

```
STACKS  Ø,,EXT
```

Force dynamic allocation of 256 word page for main stack, twelve word PSD stack, main stack to be extensible.

DEFINE A RESIDENT TASK'S MAXIMUM CONCURRENT USE OF I/O RESOURCES

A task can queue up many I/O operations concurrently to like or different I/O devices. As an aid in conserving the privileged memory areas required to permit this concurrent usage of I/O device queues, the IOPERATIONS statement lets a specific task estimate the worst case for queueing of concurrent I/O operations. If the number of operations actually queued exceeds this number, the result will only be slower I/O, causing the task to relinquish until some I/O operation completes and the I/O node for that operation becomes free for re-use for subsequent I/O operations.

SYNTAX

```
                          1        2
     IOPERA[TIONS]   [numpri] [numglo]
```

[numpri] - Parameter 1 is optional. It specifies the number of concurrent operations that the task can queue to any number of devices without using global I/O nodes. The default value is 2.

[numglo] - Parameter 2 is optional. It specifies the maximum number of global I/O nodes the task can use. When the task runs out of private I/O nodes, the task can allocate up to "numglo" number of I/O nodes from the global pool (if the nodes are available) for temporary use. The default value is zero.

EXAMPLES

```
     IOPERATIONS  10,3
```

Allow space for the queueing of up to 10 concurrent input/output operations using private I/O nodes. Use a maximum of 3 global I/O nodes (if available) when all I/O private nodes are used.

```
     IOPERA
```

Allow space for the queueing of up to 2 concurrent input/output operations. No global nodes can be used.

POPTIO

SPECIFY EACH RESIDENT TASK'S INITIAL PROGRAM OPTIONS

The POPTIONS statement is optional. It specifies the resident task's initial program options. Every task has a pseudo sense-switch register maintained uniquely for it in its TCB. This 16-bit data word can be preset to an initial value at system COLD START. No system functions use these "sense switches". The user may include coding in the program to alter its run-time behavior according to the value of bits in this word. The system operator may individually set the state of these bits with the POPTION operator directive. Tasks can also set/reset each other's program options (see GETPOP, SETPOP executive services). Each option bit is reset (=Ø) if its keyword is not specified here. The keyword used for each bit is its single digit hexadecimal bit number preceded by a P (for example, PØ, P1, ..., P9, PA, ..., PF).

SYNTAX

```
                           1     2         16
     POPTIO[NS]         [popØ] [pop1]...[popF]
```

```
[popØ]               - Parameters 1, 2, ..., 16 are optional. They
[pop1]                 specify the keyword for one of the sixteen
 ...                   bits in this pseudo sense-switch register.
 ...                   When the keyword is present, that bit in the
[popF]                 register is set. All bits that are not
                       specified are reset. The default is all bits
                       reset.
```

EXAMPLES

```
     POPTIONS   PØ,P1,P2,P3,P4,P5,P6,P7
          Define the program options as Bits Ø-7 set and Bits 8-F
          reset.

     POPTIO
          Define the program options as all bits reset.
```

DEFINING RESIDENT TASK'S INITIAL SYSTEM OPTIONS

Every task has a second option register, similar to the pseudo-sense switch register specified by the POPTIONS statement. This option register is reserved for standard system related functions that alter the run-time behavior of the task in the system. For example, a task may be placed dynamically under the control of the optional resident "DEBUG" system, made to automatically be dumped if it should abort, or made to enter the HOLD state whenever the loader completes the loading of an overlay on its behalf. Each system option is reset if its keyword name is not specified. The keyword used for each bit is its single digit hexadecimal bit number preceded by the letter U (for example, UØ, U1, ..., U9, UA, ..., UF) but some bits have alternate symbolic names (for example; LO[AD], BO, SC[RATCH], HO[LD], MA[P], GO, AO, HL, DU[MP]) which more conveniently describe this function. See Appendix D for the function of these names.

SYNTAX

```
                          1      2        16
     OPTION[S]          [optØ] [opt1]...[optF]
```

```
[optØ]                - Parameters 1, 2, ..., 16 are optional. They
[opt1]                  specify the keyword for one of the sixteen
 ...                    bits in this option register. When the
 ...                    keyword is present that particular bit in the
[optF]                  register is to be set. All bits that are
                        not specified are reset. The default is all
                        bits reset.
```

EXAMPLES

```
     OPTIONS  UØ,U1,BO,DU
          Define the system options as Bits Ø, 1, 8, and 15 set,
          and others reset.

     OPTION
          Define the system options as all bits reset.
```

DEFINE EACH VARIABLE OF EACH RESIDENT TASK

A task can specify the presence of unique named "mailboxes" in its TCB (EXT) which may be initialized with an associated 16-bit value. This value may be changed by name by the system operator or any other task wishing to use this method of inter-task communication (for example, the OC task). The statement may be repeated as necessary to specify any number of specified variables and their initial values. If no parameter is specified, exactly one "unspecified" variable (no specific name) will be added to the TCB as the first or only variable. Other variables may be specified with names and initial values with other statements which follow.

SYNTAX

```
                         1      2
     VARIAB[LES]        [vname value],...
                        [count]
```

[vname]
[count] - Parameter 1 is optional. It specifies either the name of the variable or "mailbox" (1- to 3- CAN-codable characters) or the number of blank entries to set aside with no names or values. If a name is given, it must be unique to this task. There is no default for either possible parameter.

[value] - Parameter 2 is required if Parameter 1 specifies a variable name. Parameter 2 specifies the value of the variable name given as Parameter 1. The value may be given in decimal, hexadecimal (preceded by a # character), CAN-code (preceded by a @ character), or ASCII (enclosed in " characters).

[vname] - Parameters 3, 5, ..., 15 are optional and, if present, are the same as Parameter 1.

[value] - Parameters 4, 6, ..., 16 are optional and, if present, are the same as Parameter 2.

EXAMPLES

```
     VARIABLE   5
          Reserve space for five blank variables.

     VARIAB   REM,Ø,STA,Ø
          Reserve  space  for  two  variables and set @REM = Ø and
          @STA = Ø initially.

     VARIABLES   SAM,"AM",JOE,#1234,ED,@ED,BIL,13
          Reserve  space  for  four variables and set @SAM = "AM",
          @JOE = #1234, @ED = @ED, and @BIL = 13 initially.
```

LOGFIL

DEFINE EACH LOGICAL FILE OF EACH RESIDENT TASK

A task can specify one or more unique logical file names to be associated with it and these logical files can be bound to logical devices or to other logical file names. The first assignment (Parameter 2) is the one used by the task for any I/O operation that references the logical file name at run-time. The second assignment (Parameter 3) is called the default assignment and is not used at run-time unless invoked by the ASSIGN service to replace the normal assignment. If Parameter 3 is not given, the statement uses the value of Parameter 2.

SYNTAX

```
                              1             2              3
     LOGFIL[E]        logfilename {logdev name} [logdev name]
                                  {logfilename} [logfilename]
```

logfilename - Parameter 1 is the logical file name (1- to 3-CAN-code characters), which must be unique for all the logical files for this task.

{logdev name}
{logfilename} - Parameter 2 specifies the normal mapping or assignment of logical file to physical device. This parameter may specify another logical file name in which case the ultimate physical device will be that specified by the other logical file name or a logical device name which is just the name of a physical device.

[logdev name]
[logfilename] - Parameter 3 is optional. It specifies the same type of information as Parameter 2, above. The default is the value of Parameter 2.

EXAMPLES

```
     LOGFILE   LM,QLM
          Define logical file name LM as logical device QLM (both
          normal and default).

     LOGFIL   CR,CR
          Define logical file name CR as logical device CR (both
          normal and default).

     LOGFILE   XYZ,TY,CR
          Define logical file name XYZ as logical device TY
          (normal) and CR (default).
```

DEFINE ADDITIONAL VACANT LOGICAL FILES FOR RESIDENT TASK

This form of the LOGFIL directive allows a task to have a number of vacant logical files that initially have no name and no assignment. The task can use these vacant logical files to bind specific logical devices or logical files at run-time.

SYNTAX

```
                    1
     LOGFIL[ES]     number
```

```
number              - Parameter 1 is the number  of  vacant  LOGFILE
                      entries  to  generate.   All values are set to
                      zero (Ø).
```

EXAMPLES

```
     LOGFILES   5
          Define five vacant entries.

     LOGFIL   3
          Define three vacant entries.
```

FCBDEF

DEFINE THE NUMBER OF FILE MANAGER FILES WHICH MAY BE OPEN CONCURRENTLY

The FCBDEFINE statement is optional. It specifies the number of File Manager files that may be concurrently open. The system does not assume that any particular number of real files are needed and so the system generation process does not automatically provide space for real files -- preferring to leave the choice to the system user. This statement does not, in fact, define File Control Blocks (FCB) since the File Manager creates/destroys FCB's dynamically, but instead creates File Assign Table (FAT)/Logical Device Table (LDT) pairs. Then, when a real file is opened, the FAT, the LDT, and the FCB are linked so that the file is compatible with both the File Manager and BIOS. This statement can only be used for resident tasks.

SYNTAX

```
                      1
     FCBDEF[INE]    number
```

number - Parameter 1 specifies the number of concurrent open File Manager real files that are to be allowed. There is no default value and a number must be given if the statement is used.

NOTE: In order to reduce the SIZE of a TCB for a task using the File Manager, do not specify this statement. Instead, specify vacant or named BIOS FAT's (LOGFILE) and increase the number of transient disc LDT's (TLPDEFINE).

EXAMPLES (only one usage permitted per task)

```
     FCBDEFINE   5
          Define space for five real files.

     FCBDEF   1
          Define space for one real file.

     FCBDEFINE   8
          Define space for eight real files.
```

SPECIFY SYSTEM PAGE REQUEST EXTENSION

The SYSPAGES statement is required if the task will be dynamically requesting system virtual pages. All unused system page definitions are contained in a LIFO push stack in MAP Ø. These pages are available for allocation to any task. An extension to the TCB is generated to keep a record of all pages dynamically requested so as not to lose track of these pages once they are allocated.

SYNTAX

```
                     1
     SYSPAG[ES]      numpages
```

numpages - Parameter 1 specifies the maximum number of system pages that this task will ever have concurrently allocated.

EXAMPLES (only one usage permitted per task)

```
     SYSPAGES   1Ø
          Define a ten entry page extension.

     SYSPAG   4
          Define a four entry page extension.
```

USER

ADD A USER EXTENSION TO THE EXT OF THIS TASK

The USER statement causes a user extension of "nwords" to be added to the EXT of the task (this extension is not used by MAX IV). The user area is pointed to by EXTUSR in the EXT of the task. Word Ø of the user extension is initialized to the number of words in the extension, all other words are initialized to Ø. If this statement is not specified no user extension is generated.

SYNTAX

```
                        1
     USER             nwords
```

nwords - Parameter 1 specifies the number of words to be defined in the user extension.

EXAMPLES

```
     USER   35
          Define a user extension of 35 words

     USER   512
          Define a user extension of 512 words
```

SPECIFY PAGE SHARING EXTENSION

The PAGESHARE statement is required if the system page sharing services are used by the task. When shared pages are requested by a task, the region is recorded in the task TCB to preserve the definition until the task exits or the page is extracted from the task's addressing space.

SYNTAX

```
                        1
     PAGESH[ARE]    numregions
```

numregions - Parameter 1 specifies the maximum number of shared regions that this task will ever have concurrently inserted.

EXAMPLES (only one usage permitted per task)

```
     PAGESHARE   5
          Define a five-entry share extension.

     PAGESH   2
          Define a two-entry share extension.
```

DECLARE END OF A RESIDENT TASK'S DEFINITION

The FINTASK statement specifies the end of a series of Task Control Block (TCB) defining statements which was started by the TASK statement with the same task name as that given as Parameter 1 on this statement. Upon encountering the FINTASK statement, the actual body of the TCB is generated according to the previously specified parameters of the other task defining statements. The indicated default values are used for parameters not specified. Following generation, this TCB becomes part of the permanent CPUQ linked list in the system nucleus and it contains external references to the task program which must be satisfied.

The following gives the format of the external names that are declared and that must be defined within the object module of the task body in order to satisfy the LINKER/LOADER. This object module may be part of the System Elements Library, the User Library, or attached as an assembly subsequent to this assembly. FORTRAN-coded task programs may have these names bound to the task by the Link Editor (EDI):

```
C$name - start of task body
T$name - transfer address of task program
E$name - end of task body
```

After this FINTASK statement is processed, another series of task defining commands (TASK, ..., FINTASK) may be given to define another permanently resident system task.

SYNTAX

```
                     1
FINTAS[K]          tname
```

tname - Parameter 1 is the name of the task (1- to 3-CAN-codable characters). It must match Parameter 1 given on the TASK statement which began this series of task defining statements.

EXAMPLES (only one usage permitted per task)

FINTASK X
Define end of Exceptional Condition (X) task and generate TCB.

FINTAS OC
Define end of Operator Communications (OC) task and generate TCB.

DECLARE END OF THIRD SYSTEM-GENERATION ASSEMBLY

The ENDTASKS statement is required. It finalizes the task-related data structures and terminates the Macro Assembly (outputs an END statement) for this third and possibly last section of the system generation source file. The resulting incomplete main program defines one-third (or one-fourth) of the standard system data structures. This assembly may be followed by other Macro Assemblies that are defined in PART 4 or by any number of custom, user-coded task programs. If the assembled programs are to be unconditionally linked as part of the system nucleus all sections of the SYSGEN source file are assembled before an End-of-File (EOF) mark is written to the Binary Output (BO) file. The user may choose to always keep the assemblies of custom elements contiguous with the system generation source file. If the elements are to be linked only if they have been externally referenced by an earlier assembly, then the custom elements can be placed in object form on a User Library file. This library would be considered as the first searchable library when the system parts are link-edited.

SYNTAX

```
ENDTAS[KS]
```

EXAMPLES (only one usage permitted per assembly)

```
ENDTASKS
     Define end of PART 3 assembly.

ENDTAS
     Define end of PART 3 assembly.
```

2.5 PART 4 - FILE MANAGER STRUCTURE BLOCK

All of the statements described below are optional and, if present, must immediately follow the statements described in PART 3. A separate Macro Assembly step is used so that memory requirements to effect a system generation are reduced to an acceptable level.

The File Manager is provided as an optional extension of a standard MAX IV Operating System, which may be included in a configuration by supplying the appropriate SYSGEN statements. Prior to describing the SYSGEN statements in detail, it is necessary to understand the manner in which the program elements which constitute the File Manager are organized to be configured at SYSGEN.

The File Manager is composed of the following groups of program elements:

- File Manager Nucleus.
- File Manager Optional Packages.

The File Manager Nucleus consists of those File Manager program elements that must be permanently resident in the Operating System when the File Manager is configured.

File Manager Optional Packages consist of groups of program elements that support operation of the optional features of the File Manager:

- The Security Package - provides support of the definition, maintenance, and usage operations required to fully utilize the UID/Permissions scheme of the File Manager.
- The Audit Operations Package - provides support of the audit trail generation operations.
- The Catalog Package - provides the operations which acquire and search the File Catalog.

In addition to configuring the program elements of the File Manager, the installation has the option of specifying the following items of information which - if supplied - will be used by the File Manager services:

- Maximum File Name Length.
- Transport Class Names.
- Descriptor Default Values.
- Catalog of First Level Files.
- Volume Disposition Actions.

- File Disposition Actions.

- Access-Method-Dependent Data File CLOSE Processing Routines.

- Operator Optional Message Suppression.

The set of statements that configure the File Manager for operation in a particular operating system environment are specified in a separate SYSGEN assembly, which follows the normal Operating System SYSGEN. In the assembly process, any statements that contain errors generate assembly errors and diagnostic messages are output. If statement errors occur, the bulk of the File Manager SYSGEN output is suppressed. The File Manager SYSGEN output is - with the exception of the list of any default values for file descriptors and the specification of REX services - produced in an order that is independent of the sequence in which statements are input.

The File Manager's Exceptional Condition Task (FMX) must be configured in the system as a resident task by specifying the appropriate statement when defining resident tasks in PART 3 of the SYSGEN source file.

The following sections describe the SYSGEN statements that are used to configure the File Manager and define optional File Manager information.

TTL

LISTING TITLE - BEGIN SEPARATE ASSEMBLY

The TTL statement is a standard Macro Assembler statement. Use of a TTL statement is optional. It can contain any message. It ensures that the SYSGEN listing starts on a whole page boundary -- and that the title specified appears on each page of the listing.

SYNTAX

```
                           1
     TTL                heading
```

heading - Parameter 1 indicates the heading that appears at the top of each page of this part of the SYSGEN listing.

EXAMPLE

```
     TTL   MAX IV: FMSBLK
```

DEFINE THE FILE MANAGER MACRO LANGUAGE

Before the macro language that defines the MAX IV File Manager structures can be used, the prototypes that define the language must be inserted from an external file. This standard Macro Assembler statement performs this function and MUST be present.

SYNTAX

```
                        1     2
     INS*              fname FMSMAC
```

fname - Parameter 1 specifies the logical file name where the macro prototypes are cataloged. The MAX IV standard for this file is FMM which is usually the same file as MC, the Common Macro Library. This file name must be assigned to a real file or device before it can be used.

FMSMAC - Parameter 2 is the name of the particular macro group used to define this macro language.

EXAMPLE (only one usage permitted per assembly)

```
     INS*   MC,FMSMAC
          Define macro language.
```

All the SYSGEN statements in PART 4 are defined by the MAX IV File Manager Structure Prototypes. These definitions are included in the assembly by using this statement.

NAME OF PROGRAM

This standard Macro Assembler statement is optional and assigns a name to the program being assembled. This statement must follow the INS statement in order to override the standard default provided by the macro file inserted for this block of the SYSGEN.

SYNTAX

```
                      1
     PGM             name
```

name - A valid symbolic name as defined in the MODCOMP Assemblers, Language Reference Manual. FM.UCB is the default name for the File Manager Structure Block.

SPECIFY THE MODE OF RESIDENT FILE MANAGER SERVICES

The RESSERVICE statement (required) specifies the mode by which File Manager REX services are made resident.

This statement must follow the INS statement above. All File Manager REX services must be resident.

SYNTAX

```
                        1
    RESSER[VICE]    servicesmode
```

servicesmode - The parameter specifies the mode by which services are to be permanently resident. The value may be specified in one of the following forms:

- ALL - to indicate all services are to be MAP Ø resident.
- MAPPED - to indicate all services will be non-MAP Ø resident.

NOTE: This must be the first configuration statement specified in the File Manager portion of SYSGEN as it INSERTS default values of assembly symbols.

EXAMPLE (only one usage permitted per assembly)

```
     RESSERVICE   ALL
          Specify all services resident
```

SPECIFY CONFIGURATION PARAMETERS

The FILEMANAGER statement is optional. It indicates that particular parameters of the configuration of the File Manager are to be altered.

The allowable values specify that optional operator message output is to be suppressed. The parameters are order sensitive.

SYNTAX

```
                          1       2
    FILEMA[NAGER]     [[NOARC] [NODES]]
```

[NOARC] - Parameter 1 specifies that the area control lockout operator messages are to be suppressed.

[NODES] - Parameter 2 specifies that the temporary file destroy operator messages are to be suppressed.

EXAMPLES

FILEMA
: Messages are displayed for area control lockout and when temporary files are destroyed.

FILEMAN NOARC
: Messages are displayed when temporary files are destroyed but area control lockout messages are suppressed.

SPECIFY MAXIMUM FILE NAME SIZES

The MAXNAME statement permits the installation to define the maximum number of bytes which may be supplied to constitute:

- The complete name of any file.
- The simple name of a first level file.

SYNTAX

```
                    1               2
     MAXNAM[E]      [compoundname] [simplename]
```

[compoundname] - Parameter 1 specifies the maximum number of bytes that may be used to define the complete name of any file in the system, including delimiters. The parameter's value is specified as a number in the range 1 to 65. The default value is 16.

[simplename] - Parameter 2 specifies the maximum number of bytes that may be supplied to define the simple name of any first level file in the system, including the name's delimiter. The parameter's value is specified as a number in the range 1 to 65. The default value is 6.

The value specified for "compoundname" is used by the File Manager:

- As a limit to the size of a complete file name.
- To reserve storage for file names in the File Control Block (FCB).

The following aspects of File Manager and its data structures should be considered in specifying these values:

- The File Manager's syntax delimits every simple file name with an explicit delimiter or an ASCII "space" character.
- Because file names can be of variable length, the File Manager data structures force a supplied simple name to be altered such that the number of bytes constituting the name is inserted in a byte and that byte is prefixed to the file name supplied.

- o While the byte that originally delimited the name is discarded, the byte containing the name length replaces it. Consequently, from the File Manager's standpoint - as reflected in data structures - the length of a simple file name in bytes always includes the delimiter.

- o Every simple name is forced to occupy an integral number of words.

- o As a consequence of the above, the storage required for a simple name whose length - including the delimiter - is "n" bytes always consists of "n+1/2" words.

The length of first level file names affects the amount of time required to search a directory on a randomly accessible volume. System performance can be improved by maximizing the number of entries that can be stored in a single directory segment. The second parameter permits the user to control name entry size at the first level so as to increase the information density per segment and minimize directory search overheads.

EXAMPLES (only one usage permitted per assembly)

```
MAXNAME   65,8
     Total size = 65 bytes, first level = 8 bytes

MAXNAM   16,4
     Maximum file name length = 16 bytes first level max file
     name = 4 bytes

MAXNAME   32,8
     Total = 32 bytes; first level = 8 bytes
```

DEFINE TRANSPORT CLASS TRANSPORT NAMES

The TCLASS statement permits the user to define a class of transports and the names of those transports that belong to the class.

SYNTAX

```
                          1        2       3
     TCLASS            classname tname [tname],...
```

classname - Parameter 1 defines the name of the tranport class. Its value is specified by 1- to 3- CAN-code characters.

tname - Parameter 2, 3, etcetera, specifies, in each occurrence, the name of a configured logical transport which is a member of the class. The values is specified by 1- to 3- CAN-code characters.

One statement must be used for each class defined and the names of all of the logical transports belonging to the class must appear in a single statement. As many statements as necessary to define all desired transport classes may be used.

If no statements of this type appear, no transport class names are defined in the system.

EXAMPLES (only one usage permitted per assembly)

```
     TCLASS    MHD,MØ,M1
          Class = MHD, transports = MØ, M1.

     TCLASS    MT,MT1,MT2,MT3
          Class = MT, trans = MT1,MT2,MT3.

     TCLASS    FH,FHØ
          Class = FH, transport = FHØ.
```

FDLDEF

DEFINE THE FDL OF FILE DESCRIPTORS' DEFAULT VALUES

The FDLDEFAULTS statement indicates that the statements to follow will define a set of default values for file descriptors. The FDLDEF statement must be followed by the file descriptor default values desired.

These statements must define a valid, FDL, including an ENDFDL or EDL. Failure to include an ENDFDL or EDL causes an error message to be output.

If this statement does not appear, no installation-standard default values exist.

SYNTAX

```
     FDLDEF[AULTS]
```

EXAMPLES (only one usage permitted per assembly)

```
     FDLDEFAULTS
     TNA    LØ
     VNA    SYSVOL
     GEO    4
     RSL    128
     FSG    1-TRACK
     EDL
```

The file descriptor default values for this system are defined as follows. The default transport name is LØ. The default volume name is SYSVOL. The default geometry is 4. The default record size limit is 128. The default file space granularity is 1 track. The default descriptor list is terminated by an EDL or an ENDFDL.

SPECIFY DISPOSITION ACTIONS

The DISPOSITION statement specifies the number of installation-supplied volume and/or file disposition action routines which the installation wishes included in the system. The statement causes the routines to be included in the system and generates data structures defining the entry points of these routines.

SYNTAX

```
                       1              2
   DISPOS[ITION]    [fileactions] [volumeactions]
```

[fileactions] - Parameter 1 defines the number of file disposition action routines that the user will supply. The value is specified as a number in the range 1 to 255.

[volumeactions] - Parameter 2 defines the number of volume disposition action routines that the user will supply. The value is specified as a number in the range 1 to 255.

The presence of this statement causes one or two tables to be generated. Each table consists of a number of words equal to the number of actions specified. Assembly statements are generated so that each word of the tableis filled - when the system is link-edited - with the entry point address of a routine named as follows:

- o U$FDxx - for file disposition action routines.
- o U$VDxx - for volume disposition action routines, where "xx" is replaced by the hexadecimal representation of the decimal number of a disposition action code.

If this statement does not appear, no disposition action routines exist.

EXAMPLES (only one usage permitted per assembly)

DISPOSITION 2,1
Define file disposition routines U$FDØ1-U$FDØ2 and volume disposition routine U$VDØ1 as external.

DISPOS 4
Define file disposition routines U$FDØ1-U$FDØ4 as external but no volume disposition routines.

DISPOSITION ,2
Define no file disposition routines and volume disposition routines U$VDØ1-U$VDØ2 as external.

FMCATA

SPECIFY FILE MANAGER CATALOG

The FMCATALOG statement specifies that the File Manager's Catalog feature will be used in the system and causes the File Manager program elements that implement the feature to be included in the system.

SYNTAX

```
                        1             2              3
     FMCATA[LOG]    [defaultvol] [catalogfile] [resbufsize]
```

[defaultvol] - Parameter 1 defines the name of the volume to be used if no catalog file exists or if a search of the catalog does not resolve a volume name. The value is specified as a Volume Name descriptor. If not specified, no such default exists.

[catalogfile] - Parameter 2 defines the name of the file that contains the catalog. The value is specified as 1- to 3- CAN-code characters. If this parameter is not supplied, no file-resident catalog is assumed to exist.

[resbufsize] - Parameter 3 defines the number of words to be used as storage for a memory-resident portion of the catalog. The value is specified as a number. If "Ø" is specified or the parameter is not supplied, no storage is reserved and the catalog file is searched from the beginning each time such a search is necessary.

If neither "defaultvol" or "catalogfile" is specified, an error message is output. If no "catalogfile" is specified, any specification of resident buffer size is ignored.

If this statement does not appear, those File Manager elements that provide the related services are not included in the system.

EXAMPLES (only one usage permitted per assembly)

```
     FMCATALOG   SCRVOL
          Default volume name = SCRVOL.

     FMCATALOG   ,CDF
          Catalog logical file name = CDF.

     FMCATA    ,CDF,2ØØ
          Catalog logical file name = CDF and resident directory
          buffer size = 2ØØ words.
```

SPECIFY ACCESS METHODS' CLOSE PROCESSING ROUTINES

The CLOSEMETHODS statement causes linkages to the access methods' CLOSE processing routines to be constructed for use by the File Manager CLOSE service.

SYNTAX

```
                          1                 2
   CLOSEM[ETHODS]    [systemsupplied] [usersupplied]
```

[systemsupplied] - Parameter 1 specifies the number of system-supplied access methods in use in the system. The value is specified as a number in the range Ø to 127.

[usersupplied] - Parameter 2 specifies the number of user-supplied access methods in use in the system. The value is specified as a number in the range Ø to 127.

If this statement is not specified, the only special processing action that will be performed by the File Manager when closing a data file is to relinquish until any Quick Return I/O to the file has been completed by the Basic I/O System.

The presence of this statement causes one or two tables to be generated. Each table consists of a number of words equal to the number of methods specified plus one word to indicate the number of methods. Assembly statements are generated so that each word of the table is filled - when the system is link-edited - with the entry point address of a routine named as follows:

- S$CLxx - for system-supplied access methods.
- U$CLxx - for user-supplied access methods, where "xx" is replaced by the hexadecimal digits representing the access method code number.

NOTE: Unspecified access methods will execute the routine FM$QIO to quiesce any Quick-Return I/O.

EXAMPLES (only one usage permitted per assembly)

CLOSEMETHODS 2
Define routines S$CLØ1-S$CLØ2 external close file access methods Ø1/Ø2.

CLOSEM ,1
Define routine U$CLØ1 as external user-supplied close method Ø1.

CLOSEMETH 3,1
Define routine S$CLØ1-S$CLØ3 as system-supported close method and U$CLØ1 as user-supplied close method.

FMSECU

INCLUDE FILE SECURITY PACKAGE

The FMSECURITY statement causes the program module that implements the file security scheme of the File Manager to be included in the system.

SYNTAX

```
                          1           2         3
     FMSECU[RITY]    [MasterUID] [NOOWN] [DEFIAM]
```

[MasterUID] - Parameter 1 specifies the value designated to be the optional Master UID for the installation. The value may be specified in any of the following forms:

- o Specify up to six bytes where the first character must be alpha
- o Specify up to nine CAN-code characters preceded by "@"
- o Specify up to 12 hexadecimal characters preceded by "#"
- o Specify up to nine decimal digits

[NOOWN] - Parameter 2 specifies that the File/Volume Owner descriptor is to be ignored during UID/permission processing. A file or volume is considered secured when the keyword NOOWN is specified, if it has a UID list or a permanent UID Permission specified.

[DEFIAM] - Parameter 3 specifies that the IAM descriptor is to automatically default to zero (universal) if it is not specified or if its value is not matched with a defined UID.

If this statement does not appear, all users of the File Manager always have all permissions and the file system is totally unprotected.

EXAMPLES (only one usage permitted per assembly)

```
     FMSECURITY   MUID
          Master UID = MUID plus security.

     FMSECU   FRED
          No master UID defined for system.
```

INCLUDE AUDIT PACKAGE

The FMAUDITOR statement causes the program module that provides auditing services to be included in the system and defines:

- The names of the files into which audit entries are to be placed.

- The subsequent processing actions to be taken when the audit file has been filled with entries.

SYNTAX

```
                        1            2
     FMAUDI[TOR]    auditfile [altaudfile]

                        3             4              5
                    [endaction1] [endaction2] [endaction3]
```

auditfile - Parameter 1 defines the name of the file into which audit trail entries are to be placed. The value of this parameter is specified as 1- to 3- CAN-code characters.

[altaudfile] - Parameter 2 defines the name of the file that is used as the alternate for the above file if endaction1 "AF" is specified. If this Parameter 2 is omitted it defaults to the CAN-code value NO.

[endaction1] - Parameter 3 defines the action to be taken when the audit package reaches end-of-medium on the file with the name specified in Parameter 1. The action to be taken is selected by one of the following keywords:

CY - When EOM is reached on the audit file reset the FPI of the next sector to be written to sector zero of the file specified as Parameter 1. CY is the default.

AF - When EOM is reached on the audit file make the current audit filename equal to the alternate audit filename and set the alternate audit filename equal to the previous value of the audit filename.

[endaction2] - Parameter 4 defines the audit trail processing that is to be performed following the action specified by "endaction1". The action to be performed is selected by specifying one or the other of the following keywords:

RA - To indicate that audit trail entry generation is to be resumed. RA is the default.

SA - To indicate that no further audit trail entries are to be generated.

[endaction3] - Parameter 5 specifies whether or not the system operator is to be notified of the "endaction2" when it is taken. The action to be taken is selected by specifying one or the other of the following keywords:

NS - Notify the system operator using the CO file.

DN - Do not notify the system operator. DN is the default.

Points at which audit trail entries are to be generated must be specified through use of the "AUDITPOINT" statement described on the following page if auditing operations are to be initiated as soon as the system starts up.

If this statement does not appear, the File Manager elements that support audit operations are not included in the system.

EXAMPLES (only one usage permitted per assembly)

```
FMAUDITOR   SAF,AAF,AF,RA,NS
```

Request presence of audit package, set initial audit file name = SAF, set alternate audit file name = AAF, switch to AAF when SAF fills and conversely, resume auditing after switch, and notify operator of switch.

```
FMAUDI   FAF
```

Request presence of audit package, set initial audit file name = FAF with no alternate file, treat file as circular without operator notification.

DEFINE AUDITING POINTS

The AUDITPOINT statement defines the points in File Manager service execution at which audit trail entries are generated as soon as the system starts.

SYNTAX

```
                     1            2              3
     AUDITP[OINT]    serviceid [auditpoint] [auditpoint],...
```

serviceid - Parameter 1 specifies the service(s) for which auditing points are to be defined. The parameter's value is specified as one of the following keywords to indicate the selected service(s):

MOU - To select the MOUNT service.
LAB - To select the LABEL service.
REL - To select the RELABEL service.
CRE - To select the CREATE service.
REF - To select the REFILE service.
OPE - To select the OPEN service.
EXP - To select the EXPAND service.
CON - To select the CONTRACT service.
EXT - To select the EXTEND service.
END - To select the ENDFILE service.
CLO - To select the CLOSE service.
DES - To select the DESTROY service.
DIS - To select the DISMOUNT service.
FIN - To select the FINFORMATION service.
FIL - To select the FILEDESCRIBE service.
ALL - To select all services.

[auditpoint] - Parameter 2 through 7 specifies the point(s) during service execution at which an audit trail entry is to be generated. An individual point is selected by supplying one of the following keywords:

EN - To audit service entry.
AT - To audit service termination under any condition.
ET - To audit service termination under error conditions only.
SV - To audit security violations only.
AS - To audit all steps.

If no statement of this type appears, no auditing operations are inititated until audit points are defined through use of the appropriate Operator Directive (APT).

EXAMPLES

```
     AUDITPOINT   ALL,AS
          Audit all services at all points.

     AUDITP   ALL,ET
          Audit all services on error terminate.

     AUDITP   CRE,SV
     AUDITP   DES,SV
          Audit CREATE and DESTROY services on security violations
          only.
```

DEFINE END OF FILE MANAGER CONFIGURATION

The FMEND is a required statement. It finalizes the File Manager data structures and terminates the Macro Assembly (outputs an END statement) for this section of the system generation statement source file. The resulting incomplete main program defines one-fourth of the standard system data structures and may optionally be followed by another Macro Assembly of custom user-coded task programs.

If the only statements given for this Part are this statement and the INS statement (and optionally the TTL statement since it generates no code), then the minimum File Manager configuration is generated. The configuration has the following characteristics:

- o All services are provided in MAP Ø resident form only.
- o Maximum length of a complete file name is 16 bytes.
- o Maximum length of a first level file name is 6 bytes.
- o No Transport Class Names are defined.
- o No list of default values for file descriptors exist.
- o No file or volume disposition action routines are recognized.
  No Catalog is recognized.
- o No access methods closing routines are recognized.
- o The file system is totally unprotected.
- o No auditing operations are permitted.

If errors have been generated by previous statements, an error message follows the appearance of this statement and no further output appears. Otherwise, the balance of the File Manager SYSGEN output appears.

SYNTAX

```
     FMEND
```

EXAMPLE (only one usage permitted per assembly)

```
     FMEND
          Define end of this phase.
```

CHAPTER 3
TRANSIENT PARTITION DEFINITION

The statements described below MUST be given independently of any other phase of the system generation process since the structures generated by this part are designed to be actively linked into the system rather than passively linked through the SYSGEN process. A separate Macro Assembly step is required since these definitions exist stand-alone. The Macro Assembly step constructs a set of Logical Device Table (LDT) data structures that are placed on the disc partition to which the Binary-Output (BO) logical file is assigned. At system COLD START or with an operator directive, these non-resident data structures are brought into memory and linked into the logical device list.

TTL

LISTING TITLE - BEGIN SEPARATE ASSEMBLY

This standard Macro Assembly statement is optional and can contain any message. Its purpose, once again, is to ensure that the SYSGEN listing starts on a whole page boundary --- and that the title specified appears on each page of the listing.

SYNTAX

```
                        1
     TTL             heading
```

heading - Parameter 1 indicates the heading that appears at the top of each page of this part of the SYSGEN listing.

EXAMPLE

```
     TTL   MAX IV: TPDBLK
```

INSERT MACRO PROTOTYPES - TO DEFINE MACRO-LANGUAGE

Before the macro language that defines the MAX IV transient partition definition structures can be used, the prototypes that define the language must be inserted from an external file. This standard Macro Language statement performs this function and MUST be present.

SYNTAX

```
                      1    2
    INS             fname IVIOS
```

fname - Parameter 1 specifies the logical file name where the macro prototypes are cataloged (the MAX IV standard for this file is MC [Common Macro Library]). This file name must be assigned to a real file or device before it can be used.

IVIOS - Parameter 2 is the name of the particular macro-group used to define this macro language. All the following statements are defined by the "MAX IV Basic I/O System Structure Prototypes".

EXAMPLE (only one usage permitted per assembly)

```
    INS   MC,IVIOS
        Define macro language.
```

TPDDEV

DEFINE CONTROLLER TYPE FOR TRANSIENT PARTITIONS

The TPDDEVICE statement is required and is used to define the record size of the device that receives the object code generated by the Macro Assembler (that is, the Binary-Output [BO] device). It is desirable at loading time (that is, when the transient partitions are brought into memory) that there be an integral number of transient partition definitions per physical record. The TPDDEVICE statement defines the record size of the BO device. This allows the DISCDEVICE statement to pad out the generated code so that this goal is met. The controller name is used as the parameter since the system supplies a standard default record size based on the controller name.

SYNTAX

```
                      1
     TPDDEV[ICE]    cname
```

cname - Parameter 1 is the controller name (generally given as "hhx") and must be chosen from one of the standard controller names as given in Appendix C.

NOTE: There must be one and only one of these statements specified per transient partition assembly.

EXAMPLES (only one usage permitted per assembly):

```
     TPDDEVICE   CD
          Choose cartridge disc controller.

     TPDDEV   FL
          Choose floppy disc controller.

     BPSXZL SET   512
          Define ZL = 512 bytes/sector.

     TPDDEVICE   ZL
          Choose special bytes/sector size.
```

DEFINE THE CHARACTERISTICS OF A [NON]STANDARD DISC CONTROLLER

The DISCCHARACTER statement is required for each transport for which transient partitions are to be defined. Parameters 3 through 9 have standard system-supplied default values as long as the first 2 characters of the controller name matches one of the standard controller names. Appendix C lists the default values.

SYNTAX

```
                           1       2      3     4     5     6
   DISCCH[ARACTER]  mapname cname [bps] [spt] [tpc] [cpt]

                        7     8     9     10
                      [CYL] [EXC] [MOV] [xcw]
                      [NOC] [NOE] [FIX]
                                  [FLO]
                                  [BUL]
                                  [LAR]
```

mapname - Parameter 1 is the transient partition transport mapping name (1- to 3- CAN-codable characters) to which these characteristics are to be applied. The name must be unique for all transports but the system defines no naming conventions.

To automatically load transient partition definitions at STARTUP the mapname must match a logical transport name in the SYSGEN. The TRA Operator Communications directive can be used to load transient partition definitions for a mapname that does not match any logical transport name in the SYSGEN. The TRA Directive can be used with any mapname.

cname - Parameter 2 is the name of the controller to which this transport is connected (refer to TPDDEVICE statement).

Parameters 3 through 10 are identical to the standard DISCCHARACTER statement. Refer to the DISCCHARACTER statement description in PART 2.

EXAMPLES

```
DISCCHARACTER  M1,CD
     Define transport M1 on controller CD.

DISCCH   X0,FL
     Define transport X0 on controller FL.

DISCCH   X1,FL
     Define transport X1 on controller FL.
```

DEFINE EACH TRANSIENT PARTITION

The DISCDEVICE statement is required for each transient partition that is defined. This statement is identical in its parameters to the DISCDEVICE statement defined in PART 2 and so the definition of the optional parameters is not repeated here. This statement specifies the name and size of a fixed, single-extent partition (contiguous number of tracks) on the referenced disc transport. These partitions will become known as "logical devices" when brought into memory and linked into the logical device list. This statement does generate a data structure known as the Logical Device Table (LDT), which is quite similar to that generated by the DISCDEVICE statement of PART 2.

SYNTAX

```
                   1     2    3     4      5     6      7       8
    DISCDE[VICE]  dname tname ntk [geom] [bps] [atk] [xutsk] [WOR]
                                                     [-infl] [NOW]
```

dname - Parameter 1 is the logical device name (1- to 3- CAN-codable characters). The name must be unique for all devices being defined in this assembly as well as for all devices present when these transient partitions are brought into memory. If such is not the case, then the partition definition which is already present in memory will remain in the logical device list and the transient partition of the same name will NOT be added to the list. Appendix C lists recommended naming conventions for these devices.

tname - Parameter 2 is the logical transport name on which this partition is being defined. The transport must already have been defined with a DISCCHARACTER statement.

ntk - Parameter 3 is the size (in number of contiguous tracks) of the partition being defined. If Parameter 6 is missing, the starting track address for this partition will be the track immediately following the last track of the partition defined by the last DISCDEVICE for this transport. The starting track for the first DISCDEVICE statement for a particular transport will be track Ø. This value may be adjusted, if desired, by a LET statement which defines the value of the label CTN%2 (where %2 = tname) to be some positive value. Positionally, a LET statement to define a new value for CTN%2 "must" appear after the DISCCHARACTER statement for the transport and before the first DISCDEVICE statement for the transport.

Parameters 4 through 8 are identical to the standard DISCDEVICE statement. Refer to the DISCDEVICE statement description in PART 2.

EXAMPLES

```
DISCDEVICE   ANA,M1,Ø13
     Define  partition  ANA of size 13 tracks on transport M1
     relative to the last.

CTNX1   SET 3
DISCDE  FMB,X1,1Ø
     Skip track Ø-2 on transport X1.  Define partition FMB of
     size  1Ø  tracks  on  transport  X1 at absolute starting
     track 3.
```

DEFINE END OF TRANSIENT PARTITION DEFINITION ASSEMBLY

The TPDEND statement is required. It finalizes the transient partition definition data structures and terminates the Macro Assembly (outputs an END statement) for the transient partition definition source file. The resulting object defines a portion of the basic input/output data structures that can be brought into memory at either COLD START or by an operator directive. This part of the system generation process is assembled independently of the other parts of the SYSGEN.

SYNTAX

```
     TPDEND
```

EXAMPLE (only one usage permitted per assembly)

```
     TPDEND
          Define end of part.
```

3.1 TRANSIENT PARTITION GENERATION

The following sequence of Job Control and Macro Assembler statements is given as an example of the type of things that can be accomplished:

```
    $JOB TPD.GENERATION   Begin job-step correctly
    $ASS BO=XXX           Tell Macro Assembler object destination
    $EXE ASM32,,BO        Give Macro Assembler desired options
    TTL            MAX IV: TPDBLK
    INS            MC,VIOMAC        Define macro language
    TPDDEVICE                       Define destination device
    DISCCHARACTER M1                Logical transport name/type
    DISCDEVICE     ANA,M1,Ø13       Transient partition
    DISCDEVICE     ANB,M1,Ø12       Transient partition
    DISCDEVICE     ANC,M1,Ø13       Transient partition
    DISCDEVICE     AND,M1,Ø12       Transient partition
       ...
       ...
       ...
    DISCDEVICE     AN5,M1,Ø18       Transient partition
    DISCCHARACTER XØ                Logical transport name/type
    DISCDEVICE     FLA,XØ,1Ø,2      Transient partition
    DISCDEVICE     FLB,XØ,1Ø,2      Transient partition
    DISCDEVICE     FLC,XØ,1Ø,2      Transient partition
    DISCDEVICE     FLD,XØ,1Ø,2      Transient partition
       ...
       ...
       ...
    DISCDEVICE     FLQ,XØ,67,2      Transient partition
    TPDEND                          Declare end of part
$$
$WEO BO                   Logically close binary output file
$EOJ                      End job-step correctly
```

3.2 SIOGEN REQUIREMENTS

The following statements comprise the SIOGEN. The SIOGEN is a separate macro assembly from the SYSGEN. The data structures generated from the SIOGEN are written to disc partition TBL, dedicated to DCS use.

3.2.1 INSERT

Before the macro language the defines the PDTs for the communications channels can be used, the prototypes that define the language must be inserted from an external file. This standard macro language statement performs this function and is required in the SIOGEN.

SYNTAX

```
        INS   fname,SIOMAC
```

fname - Parameter 1 specifies the logical file name where the macro prototypes are cataloged. The standard for this file is MC, the Common Macro Library. This file name must be assigned to a real file or device before it can be used.

SIOMAC - Parameter 2 is the name of the macro group used to define the statements in the SIOGEN.

3.2.2 SIOCHARACTERISTICS

The SIOCHARACTERISTIC statement must be specified at least once in the SIOGEN. This statement serves to set all of the defaults for the SIOCONTROLLER statements that follow. This statement can be repeated as many times as necessary in the SIOGEN to change the default parameters.

SYNTAX

```
    SIOCHA[RACTERISTIC]    cls,#gu[,csize][,parity][,stop][,echo]

                           [,delay][,baud][,bof][,sync][,mnbaud]
```

cls - Parameter 1 specifies the channel class. Enter one of the following keywords to define class of the channel:
A = Asynchronous
B = SDLC/HDLC Frame Level
C = MAXNET Asynchronous
D = MAXNET Bit Synchronous
E = MAXNET Byte Synchronous

#gu - Parameter 2 is the Controller's group number (g) and unit number (u) concatenated into one hexadecimal quantity.

[csize] - Parameter 3 is optional. It specifies the data

character size in bits. The range is 5,6,7 or 8. The default is 8. For SDLC/HDLC this parameter must be specified as "DLC" and parameters 4, 5 and 6 take on new meaning as specified in the SDLC/HDLC DCS Handler chapter.

[parity] - Parameter 4 is optional. It specifies the type of parity that is generated/checked. The keywords to select the parity are EVEN, ODD or NONE. The default is NONE. If parameter 3 is 'DLC' this parameter has no meaning.

[stop] - Parameter 5 is optional. It specifies the number of stop bits and has meaning only for asynchronous devices. The values of this parameter can be either 1, 1.5 or 2. The default is 2. If parameter 3 was specified as 'DLC", this parameter has no meaning.

[echo] - Parameter 6 is optional. It specifies the type of echo desired. The keywords used to select the echo type are ECHO or NOECHO. The default is NOECHO. If parameter 3 was specified as 'DLC' this parameter represents the secondary station address in the range of 0 to 255. The default is 0.

[delay] - Parameter 7 is optional. It specifies the delay time (in hundredths of a second) prior to each message being output. This delay time serves the same purpose as the output of NUL characters. The default is 0. This parameter has meaning only for asynchronous devices.

[baud] - Parameter 8 is optional. It specifies the line speed if internal clock is used. Asynchronous lines can support rates up to 19.2K baud. Bit Synchronous lines can support the standard baud rates up to 64K baud. MAXNET supports rates up to 19.2K baud. If external clock is desired, the keyword 'EXT' must be specified.

[bof] - Parameter 9 is optional and identifies the bottom-of-form simulation support count. This is the number of lines per page. The default is to use the hardware bottom-of-form signal. This parameter has meaning only on output devices or channels.

[sync] - Parameter 10 is optional. It specifies the sync character for synchronous channels. The sync character is specified as a hexadecimal character pair (#16 for ASCII or #32 for EBCDIC). The default is #16.

[mnbaud] - Parameter 11 must specify the baud rate for MAXNET links if EXT was specified for parameter 8.

3.2.3 SIOCONTROLLER

The SIOCONTROLLER statement must be specified for each DCS channel to be defined. There must be a one-to-one correspondence between the SIOCONTROLLER statements and the SIODEVICE statements that were specified in the SYSGEN. This statement causes a PDT to be generated. The PDTs that result from the SIOGEN are not resident in MAP Ø. Rather, this set of data structures is loaded into a MIAP by the SIO Loader task when the system is first started. The SIO Loader also downloads the PDTs for the appropriate channels into the respective LIM cards.

SYNTAX

```
SIOCON[TROLLER] dname,#chnl[,fdn][,wdog][,CL][,ridmp][,maxtc][,nbufs]
```

dname - Parameter 1 is the logical device name of the DCS channel. It must be the same device name as was specified for the corresponding SIODEVICE statement.

chnl - Parameter 2 is the subchannel number. It is a hexadecimal value with the upper byte representing the slot number and the lower byte representing the subchannel. For example, #Ø3Ø6 represents slot 3, subchannel 6. Valid slot numbers range from Ø to #F. Valid subchannel numbers range from Ø to 7.

[fdn] - Parameter 3 is optional. It is used only for full duplex channels and specifies the logical device name (dname) of the 'mate'. For simplex channels, this parameter must be left blank.

[wdog] - Parameter 4 is optional. It specifies the maximum time, in hundredths of a second, that an operation is allowed to remain at the head of the queue for this channel. The default value is Ø, which means that the operation is not timed.

[CL] - Parameter 5 is an optional keyword used to specify that an asynchronous device has a current loop interface.

[ridmp] - Parameter 6 represents the DMP number of the input channel on the satellite if connected to a 4828 on the satellite. This parameter has meaning only for those channels configured for MAXNET.

[maxtc] - Parameter 7 is an optional parameter which specifies the maximum transfer count in bytes for REX READ or WRITE operations on this channel. The default is 2Ø48 bytes. If present, this parameter must be nonzero (it may be zero only if parameter 8, nbufs, is also zero). The LIM Monitor uses the largest

value of this parameter (including defined values) and parameter 3 (bpr - bytes per record) of the SIODEVICE statement to determine the size of the LIM buffers to allocate for data transfers.

[nbufs] - Parameter 8 is an optional parameter which specifies the number of buffers that should be allocated on the LIM for use by the handler for this subchannel. Most handlers need only one (Default = 1). The bitsync handler needs a minimum of three (Default = 3).

The sum of the value of this parameter overall the subchannels SYSGENed for a LIM, plus one, makes up the total number of identically sized buffers that the LIM monitor will allocate on the LIM. How the buffer size is determined is described under parameter 7 (maxtc).

If too many large buffers are requested and there is not enough memory on the LIM, the user will get an error from the Loader task indicating that there is insufficient memory on the LIM card. To correct this, reduce the value of this parameter or parameter 7, maxtc, or parameter 3, bpr of the SIODEVICE statement.

3.2.4 ENDSIO

The ENDSIO statement is required. It serves to finalize the data structures that are generated in the SIOGEN. It also terminates the macro assembly by outputting an END statement.

```
ENDSIO
```

There are no parameters for this macro.

CHAPTER 4
SYSGEN ERROR MESSAGES

SYSGEN error messages are denoted by assembly-time error flags. These errors are sometimes created by macro expansions that intentionally generate an illegal op-code. The operand field of the error message usually contains some explanatory information. Samples of this notation in a printout are shown in EXAMPLE 4-1 and EXAMPLE 4-2.

EXAMPLE 4-1: A SYSGEN error message denoted by an assembly-time error flag

```
----------------------------------------------------------------------
|                                                                    |
|                          :           :                             |
| 5                       CPUNAME     #ID/1Ø1                        |
| 5 LS                 ?/////////////////////////////////////////////|
| 5 LO   Ø52B    ØØØØ  ? ILLEGAL   CHARACTER   IN   CPU-NAME   ???   |
|        Ø52C A ØØØØ                                                 |
| 5 LS                 ?/////////////////////////////////////////////|
| 5 C    Ø52D A 2Ø2Ø      DFC         '                              |
|        Ø52E A ØØØØ      RES         31,Ø                           |
|                          :           :                             |
|                                                                    |
----------------------------------------------------------------------
```

EXAMPLE 4-2: A SYSGEN error message denoted by a macro expansion that intentionally generates an illegal op-code

```
----------------------------------------------------------------------
|                                                                    |
|                           :           :                            |
| 3                        CALARM      #2Ø,4,255                     |
| 3 O    Ø5ØØ    ØØØØ      ERROR       ILLEGAL.INTERRUPT.LEVEL       |
|                           :           :                            |
|                                                                    |
----------------------------------------------------------------------
```

In this chapter, %N is used to designate information in the SYSGEN where N is the parameter number. An example of this follows:

EXAMPLE 4-3:

```
                         :             :
3                        INTERR        8
                         :             :
4                        INTERR        8
     Ø5Ø5 R Ø511         DFC           BACK+4          <<<---<<<
4 S                 "ERROR - INTERRUPT LEVEL 8 MULTIPLY DEFINED"
                         :             :
```

This message will be referenced in this chapter as follows:

```
  ERROR - INTERRUPT LEVEL %1 MULTIPLY DEFINED
```

The SYSGEN error messages are grouped first by the section of the SYSGEN in which they appear.

PART 1 - SYSTEM STRUCTURE BLOCK
PART 2 - BASIC I/O SYSTEM BLOCK
PART 3 - TASK STRUCTURE BLOCK
PART 4 - FILE MANAGER STRUCTURE BLOCK

The messages are presented in the alphabetical order of the statements that can cause the error. The documentation of each statement includes possible causes of the error, which should help in error recovery.

4.1 SYSTEM STRUCTURE BLOCK

CALARM STATEMENT

ER - TOO MANY COMMON ALARMS

There are more than 12 CALARM statements in the SYSGEN. There must be between Ø and 12 CALARM statements in a SYSGEN.

ERROR ILLEGAL.INTERRUPT.LEVEL

The interrupt level specified is less than #7 or it is a party-line and less than #2Ø or greater than #FF. Parameter 2 (interrupt level) specifies the true hardware interrupt level or IOIS pseudo level and as such, may be one of levels #7 through #B or #2Ø through #3F.

CPUNAME STATEMENT

? CPU-NAME TOO LONG ???

There are more than 64 characters in the parameter of the CPUNAME statement. There must be between Ø and 64 characters in the parameter.

? ILLEGAL CHARACTER IN CPU-NAME ???

A character in the parameter of the CPUNAME statement is +, =, -, $, #, or /, or translates to less than #21 or greater than #7D. Characters in the CPUNAME must exclude non-printing characters, plus, minus, hexadecimal, equal, or dollar signs, or slashes, and terminate with a comma.

DCINTERRUPT STATEMENT

ERROR ILLEGAL.INTERRUPT.LEVEL

The interrupt level specified is less than #7 or it is a party-line and less than #2Ø or greater than #FF. Parameter 2 (interrupt level) specifies the true hardware interrupt level or IOIS pseudo level and as such, may be one of levels #7 through #B or #2Ø through #3F.

ERROR - INTERRUPT LEVEL %1 MULTIPLY DEFINED

A hardware interrupt level from #7 to #B has been specified more than once. Parameter 2 of the DCINTERRUPT statement specifies the true hardware interrupt level or IOIS pseudo level and any hardware interrupt level may be used by only 1 CALARM, INTERRUPT, or DCINTERRUPT statement.

EVENTLOG STATEMENT

ERROR TOO MANY WORDS IN LOGGING NODE

Parameter 11 of the EVENTLOG statement is greater than 88. Parameter 11 of the EVENTLOG statement is the number of entry specific data words for each logging node and can not be greater than 88.

GLOBAL STATEMENT

ERROR PAGES.>.256

Parameter 2 of the GLOBAL statement is greater than 256. Parameter 2 of the GLOBAL statement is the number of actual pages to be allocated, and can not be greater than 256.

ERROR ACTUAL.PAGE.>.8191

Parameter 5 of the GLOBAL statement is greater than 8191. Parameter 5 of the GLOBAL statement is the starting point in actual memory for the defined area, and can not be greater than 8191.

ERROR XMEM.NOT.ON.8K.BOUNDARY

Parameter 6 of the GLOBAL statement is specified as XMEM and Parameter 5 of the GLOBAL statement is not an 8K memory boundary. If the global region defined is to be in extended memory (Parameter 6) the starting point of the actual memory (Parameter 5), if specified, must be on an 8K boundary.

INTERRUPT STATEMENT

ERROR ILLEGAL.INTERRUPT.LEVEL

The interrupt level specified is less than #7 or it is a party-line and less than #2Ø or greater than #FF. Parameter 2 specifies the true hardware interrupt level or IOIS pseudo level and as such, may be one of levels #7 through #B or #2Ø through #3F.

ERROR - INTERRUPT LEVEL %1 MULTIPLY DEFINED

A hardware interrupt level from #7 to #B has been specified more than once. Parameter 2 specifies the true hardware interrupt level or IOIS pseudo level and any hardware interrupt level may be used by only 1 CALARM, INTERRUPT, or DCINTERRUPT statement.

INTERTASK STATEMENT

ERROR - ILLEGAL PAGE COUNT IN INTERTASK MACRO

Parameter 1 of the INTERTASK statement is less than 1. Parameter 1 of the INTERTASK statement is the maximum number of actual memory pages that may be used at any given time by ITC, and if specified, must be greater than Ø. The default value for this parameter is 2ØØ.

ERROR - ILLEGAL WINDOW SIZE IN INTERTASK MACRO

Parameter 2 of the INTERTASK statement is less than 1 or greater than 127. The second parameter of the INTERTASK statement is the default window size for tasks not implicitly specifying a window size in using ITC, and if specified, must be between 1 and 127 inclusive. The default value for this parameter is 8.

ERROR - PARAMETER 3 NOT DEFAULTED OR EQUAL TO MAPPED

Parameter 3 of the INTERTASK statement specifies whether ITC is to be run in a map other than MAP Ø. If no parameter is specified, the default is MAP Ø. This parameter must be defaulted or specified as the keyword MAPPED. Any other value causes this error.

LOGFORMAT STATEMENT

ERROR EVENTLOG STATEMENT MISSING OR OUT OF ORDER

The EVENTLOG statement is not before the LOGFORMAT statement. The EVENTLOG statement must be present in the SYSGEN before the LOGFORMAT statement.

ERROR FORMAT PACKAGE NOT CONFIGURED ON EVENTLOG STATEMENT

Parameter 12 of the EVENTLOG statement is not @FOR. @FOR is required as the Parameter 12 of the EVENTLOG statement in order to have LOGFORMAT.

LOGPOINT STATEMENT

ERROR EVENTLOG STATEMENT MISSING OR OUT OF ORDER

The EVENTLOG statement is not before the LOGPOINT statement. The EVENTLOG statement must be present in the SYSGEN before the LOGPOINT statement.

ERROR ITC LOGPOINT NOT DEFINED

The @ITC logpoint is included in the LOGPOINT statement. @ITC is not a legal point at this time, so it may not be included.

MACHINE STATEMENT

ERROR - MODCOMP IV NOT SUPPORTED AS OF REV H.Ø

The MODCOMP IV hardware is no longer supported. All code related to the MODCOMP IV has been removed from MAX IV.

PORT STATEMENT

ERROR - TASK NAME NOT SPECIFIED IN PORT STATEMENT

Parameter 1 of the PORT statement is missing. Parameter 1 of the PORT statement is the task name and is a required parameter in the PORT statement. It must be a MAX IV task name defined in a SERVER statement.

ERROR - TOO MANY VC'S REQUESTED IN PORT STATEMENT

Parameter 4 of the PORT statement is greater than 255. Parameter 4 is the maximum number of virtual circuits that a task can have open at any one time, and can not be greater than 255.

ERROR - ILLEGAL NUMBER OF VCS REQUESTED IN PORT STATEMENT

Parameter 4 of the PORT statement specifies the maximum number of Virtual Circuits (VCs) that a task can have open at any one time. This value must be in the range of Ø to 255. If not specified the default is 4.

SERPACK STATEMENT

ERROR %1.SPECIFIES.ILLEGAL.PACKAGE

The package name specified as Parameter 1 on the SERPACK statement is not a defined package name. The package name must be one of those listed in Appendix B of this manual.

SERVER STATEMENT

ERROR ILLEGAL CHARACTER IN SERVER NAME

A character in the server name (Parameter 1) is +, =, -, $, #, or /, or translates to less than #21 or greater than #7D. Characters in the server name must exclude non-printing characters, plus, minus, hexadecimal, equal, or dollar signs, or slashes, and terminate with a comma.

ERROR NO SERVER NAME SPECIFIED

Parameter 1 of the SERVER statement is missing. Parameter 1 of the SERVER statement is the server name, which is a required parameter.

ERROR SERVER NAME TOO LONG

Parameter 1 of the SERVER statement is greater than 40 characters. Parameter 1 of the SERVER statement is the server name, and must be between 1 and 40 ASCII characters long inclusive.

ERROR TASK NAME NOT SPECIFIED IN SERVER STATEMENT

Parameter 2 of the SERVER statement is missing. Parameter 2 of the SERVER statement is the task name, which is a required parameter.

SERVICE STATEMENT

ERROR ILLEGAL.SERVICE.NUMBER.%2

Parameter 2 of the SERVICE statement is greater than 255. Parameter 2 is the number to be associated with a custom, user-coded REX service, and can not be greater than 255.

ERROR ILLEGAL.III.SERVICE.%2

Parameter 2 of the SERVICE statement is greater than 127. Parameter 2 is the number to be associated with a custom, user coded REX service, and for a compatible MAX III service can not be greater than 127.

ERROR NO.TASK.SCHEDULING.INTERRUPTS.SPECIFIED

The interrupt scheduling service has been specified and no INTERRUPT statements are specified. If the interrupt scheduling service is specified, at least 1 INTERRUPT statement must be specified.

ERROR NO.TASK.SCHEDULING.TIMERS.SPECIFIED

The timer scheduling service has been specified and no TIMER statements are specified. If the timer scheduling service is specified, at least 1 TIMER statement must be specified.

SYSBUG STATEMENT

ERROR - ADUMP DEVICE SPECIFIED WITHOUT ADUMP

Parameter 3 was specified but Parameter 2 was not ADUMP or BOTH.

ERROR - ADUMP DEVICE SUPPLIED FOR VDUMP

Parameter 3 was specified but Parameter 2 was DUMP.

ERROR - INVALID DUMP PARAMETER

Parameter 2 was specified as something other than ADUMP, DUMP, or BOTH.

SYSLOG STATEMENT

```
ERROR ERROR ERROR ERROR ERROR ERROR ERROR ERROR ERROR
USE EVENTLOG STATEMENT TO CONFIGURE EVENT LOGGING
USE LOGPOINTS STATEMENT TO CONFIGURE LOGGING POINTS
```

A SYSLOG statement was included in the SYSTEM STRUCTURE BLOCK of the SYSGEN. SYSLOG is not a legal SYSGEN statement.

SYSOPT STATEMENT

ERROR QAB NO LONGER NEEDED AS OF REV. EØØ

@QAB is specified in the SYSOPT statement. Revision EØØ removed QAB as an option for the parameter in the SYSOPT statement.

4.2 BASIC I/O SYSTEM BLOCK

BREAK STATEMENT

ERROR REFER.TO.LINEMONITOR.STATEMENT

The BREAK statement is not supported as of REV H.Ø of MAX IV. A LINEMONITOR statement is used if a Break/Ring processing task is included in the system. Refer to LINEMONITOR.

COMCONTROLLER STATEMENT

ERROR SUBCHANNEL.%3.REQUIRES.3.DIGITS

Parameter 3 of the COMCONTROLLER statement is less than 3 digits. Parameter 3 of the COMCONTROLLER statement is the controller channel number, and must be 3 digits.

ERROR INSUFFICIENT.MUX.LINKCONTROLLER.ENTRIES

Parameter 2 of the LINKCONTROLLER statement is not large enough for all of the MAXNET multiplexed link controllers present. (See the MAXNET IV System Installation Guide for further details.)

ERROR ILLEGAL.CHANNEL.NUMBER.FOR.THIS.DEVICE.CLASS

Parameter 3 of the COMCONTROLLER statement is greater than 127 and Parameter 7 is @J, @A, @B, or @C or Parameter 3 is greater than 511. Parameter 3 of the COMCONTROLLER statement is the controller channel number and must be less than 511. Parameter 7 of the COMCONTROLLER statement is the controller class, and for classes @J, @A, @B, and @C, Parameter 3 must be less than 128.

ERROR STATION.ADDRESS.MISSING

Parameter 10 of the COMCONTROLLER statement is missing, and this is a 4828 MAXNET standard link. Parameter 10 of the COMCONTROLLER statement is the secondary station address, and is a required parameter in the COMCONTROLLER statement for a 4828 MAXNET Standard Link.

ERROR ILLEGAL.PARAMETER

Parameter 10 of the COMCONTROLLER statement is greater than 255. Parameter 10 of the COMCONTROLLER statement is the secondary station address and must be less than or equal to 255.

ERROR PARAMETER.10.MUST.BE."DLC".OR.NULL.FOR.CLASS.D

Parameter 7 is D and Parameter 10 is not "DLC" or a null in the COMCONTROLLER statement. Parameter 10 must be "DLC" for a class D MAXNET device.

ERROR NUMBER.OF.TRANSPORTS.ON.CONTROLLER.%1.EXCEEDS.LIMIT

Either the number of COMDEVICE or FMTRANSPORT statements specified for this controller is greater than 1 or Parameter 7 is K, L, or M, and the number of FMTRANSPORT statement specified for this controller is greater than 2. There must be less than 3 FMTRANSPORT statements specified for any AFD controller, and no more than 1 COMDEVICE or FMTRANSPORT statement specified for any other device using the COMCONTROLLER statement.

COMDEVICE STATEMENT

ERROR FRAME.SIZE.COMDEVICE

Parameter 7 of the COMDEVICE statement is greater than 8 or less than 5. Parameter 7 of the COMDEVICE statement is the data character frame size in bits, and must be 5, 6, 7, or 8.

ERROR STOP.BITS.COMDEVICE

Parameter 9 of the COMDEVICE statement is greater than 2 or less than 1. Parameter 9 of the COMDEVICE statement is the number of stop bits, and must be either 1 or 2.

ERROR ERROR.ILLEGAL.PARAMETER.13

Parameter 13 of the COMDEVICE statement is greater than 99. Parameter 13 of the COMDEVICE statement is the sync characters in the case of SYNC controllers, and must be less than #100.

CONTROLLER STATEMENT

ERROR INSUFFICIENT.NON-MUX.LINKCONTROLLER.ENTRIES

Parameter 1 of the LINKCONTROLLER statement is not large enough for all of the MAXNET non-multiplexed link controllers present. (See the MAXNET IV System Installation Guide for further details.)

ERROR POSITIVE.CYCLE.TIMER.VALUE.REQ'D

Parameter 12 of the CONTROLLER statement is less than 1. Parameter 12 of the CONTROLLER statement is the number of ticks to be specified as a cycle timer for the 4828 MAXNET Standard Link, so it must not be Ø or a negative number.

ERROR MISSING.SECONDARY.DEVICE.ADDRESS

A PIO controller other than "WS" or "HL" has been specified, and Parameter 5 is missing in the CONTROLLER statement. Parameter 5 is a second controller address (GROUP/UNIT) which must only be included for an analog input multiplexor that uses separate I/O channels for sending addresses and receiving data.

ERROR NUMBER.OF.TRANSPORTS.ON.CONTROLLER.%1.EXCEEDS.LIMIT

o A computer to computer link ("CL") or Standard Link ("OL" and "IL") has been specified and the number of DEVICE statements specified for this controller is greater than 1. A ("CL") controller may have only 1 DEVICE statement associated with it, and since a pair of CONTROLLER statements are necessary for the Standard Link ("OL and "IL") 1 DEVICE statement only must be associated with each CONTROLLER statement.

o A plotter, printer, punched card, or paper tape punch controller has been specified, and the number of DEVICE statements specified for this controller is greater than 1. A plotter, printer, punched card, or paper tape punch controller may have only 1 DEVICE statement associated with it.

- A terminal controller has been specified and the number of DEVICE statements specified for this controller is greater than 3. A terminal controller may have no more than 3 DEVICE statements (1 terminal, 1 paper tape reader, and 1 slow speed paper tape punch) associated with it.

- A magnetic tape controller has been specified and the number of DEVICE statements specified for this controller is greater than 4. A magnetic tape controller may have no more than 4 DEVICE statements associated with it.

- A bulk memory controller has been specified and the number of transports configured for this controller is greater than 1. A bulk memory controller may have no more than 1 transport configured for it.

- A disc controller has been specified and the number of transports configured for this controller is greater than 4. A disc controller may have no more than 4 transports configured for it.

ERROR MISSING.SECONDARY.DMP.CHANNEL.NUMBER

A PIO controller other than "WS" or "HL" has been specified, and Parameter 6 is missing in the CONTROLLER statement. Parameter 6 is a second DMP channel which must only be included for an analog input multiplexor.

ERROR OPTX%1&#FF=Ø....SHOULDN'T!!!!

The configuration word for this controller is not present. The controller name used is illegal.

ERROR EOF.TABLE.ADDRESS.OF.%9.TOO.BIG

A bulk memory controller has been specified, Parameter 9 is present and not equal to @TAB, and the value of Parameter 9 is greater than the legal number of cylinders per pack. A bulk memory controller may not specify the End-of-File table to reside on a track that is not on a legal cylinder for the controller.

ERROR "PROCESS IO HANDLERS NOT SUPPORTED AS OF H.Ø"

As of MAX IV, Revision H.Ø, support of the traditional Process I/O Handlers has been eliminated. Requesting a PIO Handler using a CONTROLLER statement in a SYSGEN will generated this error.

DEVICE STATEMENT

ERROR GROUP/EXPANDER/CHANNEL.PARAMETER.MISSING

A REMAC PIO analog input or output controller has been specified, and the Parameter 6 in the DEVICE statement is missing. The GROUP/EXPANDER/CHANNEL on which the device has been placed on the REMACS terminal unit is a required parameter in the DEVICE statement. Refer to the MAX IV Communication Handlers, System Guide listed in the Preface.

DISCCHARACTER STATEMENT

ERROR ILLEGAL.BIT.NUMBER.GIVEN

An internal check has failed in the DISCCHARACTER statement. No recovery or correction is possible. The SYSGEN must be rerun from the beginning.

DISCDEVICE STATEMENT

ERROR ILLEGAL.GEOMETRY.OF.%4.ON.DISCDEVICE.%1

o Parameter 4 (GEOMETRY) is greater than the sectors per track for this DISCDEVICE.

o The specified geometry is illegal for this DISCDEVICE as follows:

  - There are 15 sectors per track and the geometry is evenly divisible by 2 or 7.

  - There are 16 sectors per track and the geometry is evenly divisible by 3 or 5.

  - There are 26 sectors per track and the geometry is evenly divisible by 5.

  - There are 64 sectors per track and the geometry is evenly divisible by 7.

Parameter 4 must be less than the number of sectors per track, and must meet certain other conditions as indicated above.

ERROR DISCDEVICE.SPACE.EXCEEDED.ON.%2

The DISCDEVICE statements for the transport specified (%2) specified more tracks than there are available. The number of tracks specified on DISCDEVICE statements may not exceed the total number of tracks available.

DISCTRANSPORT STATEMENT

ERROR "DISCTRANSPORT STATEMENT NOT SUPPORTED AS OF REV H.Ø"

The DISCTRANSPORT statement is not supported as of Revision H.Ø of MAX IV. An FMTRANSPORT statement must be used in its place. Refer to FMTRANSPORT.

ENDIOS STATEMENT

ERROR "DISC TRANSPORT DEFINED WITHOUT LOGICAL TRANSPORT"

Every physical disc transport must have at least one logical transport defined on it. This is a requirement for every Revision H.Ø or later system. See documentation for LOGTRANSPORT statement in this manual.

ERROR "PHYSICAL DISC NOT COVERED WITH LTS"

Every physical transport must be completely covered with logical transports. Check starting track definitions and logical transport sizes on LOGTRAN statements and compare against size of physical disc. Due to the way the SYSGEN macros are written, it is impossible to identify on which physical disc the problem resides.

FMTRANSPORT STATEMENT

ERROR xxx MUST BE SPECIFIED ON LOGTRAN STMNT ONLY

Parameter 4 of the FMTRANSPORT statement is unused as of Revision H.Ø of MAX IV. The value xxx, which was erroneously specified on the FMTRANSPORT statement, should be specified as Parameter 5 on the LOGTRAN statement.

LOGTRANSPORT STATEMENT

ERROR LOGTRAN.%1.DEFINED.ON.NON.DISC.TRANSPORT.%2

The capability of setting up logical transports exists only for discs. Remove statements attempting to define logical transports on non-disc physical transports.

ERROR SPACE.EXCEEDED.ON.%

The total size of all logical transports on a physical transport exceeds the size of the physical transport.

ERROR "NON CONTIGUOUS LOGICAL TRANSPORTS ON %2"

The logical transports defined of physical transport %2 are not positioned contiguously. Verify size of disc definitions and starting tracks, as specified on LOGTRAN statements for this physical transport are correct. Also check statement ordering.

4.3 TASK STRUCTURE BLOCK

BUFFER STATEMENT

ER - TASK DEF. OUT-OF-ORDER

The TASK statement is not before the BUFFER statement. The TASK statement must be present in the SYSGEN before the BUFFER statement.

ENDTASKS STATEMENT

ER - "FINTASK" MISSING

The FINTASK statement is not before the ENDTASKS statement. The FINTASK statement must be present in the SYSGEN before the ENDTASKS statement. One FINTASK statement must be present for each task to delimit the task definitions.

***** ERROR ***** "IONODE STATEMENT MISSING"

The IONODE statement is not present before the ENDTASKS statement. One IONODE statement must be present in the SYSGEN before the task definitions and ENDTASKS statement.

FCBDEFINE STATEMENT

ER - TASK DEF. OUT-OF-ORDER

The TASK statement is not before the FCBDEFINE statement. The TASK statement must be present in the SYSGEN before the FCBDEFINE statement. The FCBDEFINE statement is used to set up the file assign table for a task.

FINTASK STATEMENT

ER - TASK DEFINITION

Parameter 1 of the FINTASK statement is not equal to Parameter 1 of the last TASK statement specified. Parameter 1 of the FINTASK statement must be the name of the task in the most previous TASK statement.

ER - TASK NAME MISSING

Parameter 1 of the FINTASK statement is missing. Parameter 1 of the FINTASK statement is the name of the task in the most previous TASK statement, which is a required parameter.

IONODES STATEMENT

***** ERROR ***** "IONODE CAN BE SPECIFIED ONCE"

More than 1 IONODES statement has been specified. Only 1 IONODES statement may be present for each task.

***** ERROR ***** "NUMBER OF NODES LESS THAN ONE"

Either Parameter 1 is less than 1, Parameter 2 is less than 1, or Parameter 3 is less than 1 in the IONODES statement. The Parameters 1, 2, and 3 in the IONODES statement are quantities of nodes, and must be greater than Ø.

***** ERROR ***** "NUMBER OF NODES > 15ØØ"

Either Parameter 1 is greater than 15ØØ or Parameter 2 is greater than 15ØØ in the IONODES statement. Parameters 1 and 2 in the IONODES statement are quantities of nodes, and must be less than 15Ø1.

IOPERATIONS STATEMENT

ER - TASK DEF. OUT-OF-ORDER

The TASK statement is not before the IOPERATIONS statement. The TASK statement must be present in the SYSGEN before the IOPERATIONS statement. The IOPERATIONS statement is used to define the resources for a task.

***** ERROR ***** "NUMBER OF NODES LESS THAN ONE"

Parameter 1 of the IOPERATIONS statement is less than 1. Parameter 1 of the IOPERATIONS statement is the number of concurrent operations that can be queued without using global I/O nodes, and must be greater than Ø.

***** ERROR ***** "NUMBER OF NODES > 256"

Either Parameter 1 is greater than 256 or Parameter 2 is greater than 256 in the IOPERATIONS statement. Parameters 1 and 2 of the IOPERATIONS statement are quantities of nodes and can not be greater than 256.

***** ERROR ***** "NUMBER OF GLOBAL NODES < Ø"

Parameter 2 of the IOPERATIONS statement is less than Ø. Parameter 2 of the IOPERATIONS statement is the maximum number of global I/O nodes the task can use, and can not be less than Ø.

LOGFILE STATEMENT

ER - TASK DEF. OUT-OF-ORDER

The TASK statement is not before the LOGFILE statement. The TASK statement must be present in the SYSGEN before the LOGFILE statement. The LOGFILE statement is used to define the resources for a task.

ER - CO MUST BE 1ST FILE

This is the X task and Parameter 1 of the LOGFILE statement is not @CO. The first LOGFILE statement for the X task must be @CO.

OPTIONS STATEMENT

ER - TASK DEF. OUT-OF-ORDER

The TASK statement is not before the OPTIONS statement. The TASK statement must be present in the SYSGEN before the OPTIONS statement. The OPTIONS statement is used to define the resources for a task.

PAGESHARE STATEMENT

ER - TASK DEF. OUT-OF-ORDER

The TASK statement is not before the PAGESHARE statement. The TASK statement must be present in the SYSGEN before the PAGESHARE statement. The PAGESHARE statement is used to define the resources for a task.

POPTIONS STATEMENT

ER - TASK DEF. OUT-OF ORDER

The TASK statement is not before the POPTIONS statement. The TASK statement must be present in the SYSGEN before the POPTIONS statement. The POPTIONS statement is used to define the resources for a task.

STACKS STATEMENT

ER - TASK DEF. OUT-OF-ORDER

The TASK statement is not before the STACKS statement. The TASK statement must be present in the SYSGEN before the STACKS statement. The STACKS statement is used to define the resources for a task.

ER - STACK SIZE

Parameter 3 is not @EXT or @ROL and Parameter 1 is less than 51 in the STACKS statement. If the stack is not extensible or rollable, the stack size must be greater than 5Ø.

SYSPAGES STATEMENT

ER - TASK DEF. OUT-OF-ORDER

The TASK statement is not before the SYSPAGES statement. The TASK statement must be present in the SYSGEN before the SYSPAGES statement. The SYSPAGES statement is used to define the resources for a task.

TASK STATEMENT

***** ERROR ***** "TRANSIENTS MUST BE ENTERED BEFORE TASK"

There is no TRANSIENT statement before the TASK statement. The TRANSIENT statement must be present in the SYSGEN before the TASK statement.

ER - TASK PRIORITY

Either Parameter 2 is less than Ø, Parameter 2 is greater than 255, or Parameter 2 is less than Parameter 2 of the most previous TASK statement. Parameter 2 is the task priority, so it must be between Ø and 255 and less than the priority of the most previous task in the SYSGEN.

TRANSIENT STATEMENT

***** ERROR ***** "NUMBER OF TRANSIENTS < 1"

Parameter 1 is less than 1 in the TRANSIENT statement. Parameter 1 of the TRANSIENT statement is the maximum number of transient non-resident tasks that can ever be concurrently active, established, or scheduled in the MAX IV system being generated, and if this is included, Parameter 1 can not be less than 1.

VARIABLE STATEMENT

ER - TASK DEF. OUT-OF-ORDER

The TASK statement is not before the VARIABLE statement. The TASK statement must be present in the SYSGEN before the VARIABLE statement. The VARIABLE statement is used to define the resources for a task.

4.4 FILE MANAGER STRUCTURE BLOCK

AUDITPOINT STATEMENT

>>>ERROR: "%1 IS UNRECOGNIZED SERVICE/CONFIGURATION NAME"

Parameter 1 of the AUDITPOINT statement is not valid. Valid first parameters for the AUDITPOINT statement are REL, REF, CLO, MOU, LAB, CRE, OPE, EXP, CON, EXT, END, DES, DIS, FIN, FIL, and ALL.

>>>ERROR: "%X IS UNRECOGNIZED AUDIT POINT"

A parameter after Parameter 1 is not valid in the AUDITPOINT statement. Valid parameters for all parameters other than the first parameter are AS, SV, ET, AT, and EN.

CLOSEMETHODS STATEMENT

>>>ERROR: "NUMBER OF METHODS MUST BE SPECIFIED AS NUMBER IN RANGE Ø TO 127"

Parameter 1 of the CLOSEMETHODS statement is less than Ø or greater than 127 or Parameter 2 of the CLOSEMETHODS statement is less than Ø or greater than 127. Both Parameters 1 and 2 of the CLOSEMETHODS statement are the number of different types of access methods in use in the system, and must be numbers in the range of Ø to 127.

FMAUDITOR STATEMENT

>>>ERROR: "NO AUDIT FILE NAME SPECIFIED"

Parameter 1 of the FMAUDITOR statement is missing. Parameter 1 of the FMAUDITOR statement is the audit file name and is a required parameter.

>>>ERROR: "%3 IS UNRECOGNIZED ENDACTION1"

Parameter 3 of the FMAUDITOR statement is not valid. Parameter 3 can be AF or CY.

>>>ERROR: "%4 IS UNRECOGNIZED ENDACTION2"

Parameter 4 of the FMAUDITOR statement is not valid. Parameter 4 can be RA or SA.

>>>ERROR: "%5 IS UNRECOGNIZED ENDACTION3"

Parameter 5 of the FMAUDITOR statement is not valid. Parameter 5 can be NS or DN.

FMCATALOG STATEMENT

>>>ERROR: "NEITHER "DEFAULTVOLUME" NOR "CATALOGFILE" SPECIFIED"

Both Parameters 1 and 2 of the FMCATALOG statement are missing. Parameter 1 of the FMCATALOG statement is the default volume name and Parameter 2 is the name of the file that contains the catalog. One of these 2 parameters is required in the FMCATALOG statement.

MAXNAME STATEMENT

>>>ERROR: "MAX LENGTH COMPOUND FNA > 65 BYTES"

Parameter 1 of the MAXNAME statement is greater than 65. Parameter 1 of the MAXNAME statement is the maximum number of bytes used to define the complete name of any file in the system. This parameter's value must be in the range of 1 to 65, with a default value of 16.

RESSERVICE STATEMENT

>>>ERROR: "%X IS NOT RECOGNIZED AS SERVICE OR SERVICE CONFIGURATION NAME"

A parameter of the RESSERVICE statement is not valid. Valid parameters are REL, REF, CLO, MOU, LAB, CRE, OPE, EXP, CON, EXT, END, DES, DIS, FIN, FIL, ALL, SC1, SC2, and MAP.

TCLASS STATEMENT

>>>ERROR: "NO TRANSPORT CLASS NAME SPECIFIED"

Parameter 1 of the TCLASS statement is missing. Parameter 1 of the TCLASS statement is the name of the transport class, and is a required parameter. Its value is specified by 1 to 3 ASCII characters that can be represented in CAN-code.

>>>ERROR: "AT LEAST ONE MEMBER MUST EXIST IN CLASS"

Parameter 2 of the TCLASS statement is missing. Parameter 2 of the TCLASS statement is the name of a configured transport that is a member of the transport class. Its value is specified by 1 to 3 ASCII characters that can be represented in CAN-code.

APPENDIX A
SUMMARY: SYSGEN STATEMENTS (Alphabetical Order)

PART 1 - SYSTEM STRUCTURE BLOCK		PAGES
CALARM	Define a common alarm task scheduling interrupt	45
CLOCK	Specify clock and timer characteristics	24
CPUID	Define a CPU-ID for the system	39
CPUNAME	Define a CPU-NAME for the system	40
DCINTERRUPTS	Include a user Interrupt Routine	47
DCTIMERS	Define the number of special directly connected timers	42
DIRECTORY	Reserve space for Resident load module directories	35
ENDSYSTEM	Stop PART 1 assembly	58
EVENTLOG	Specify the existence and configuration of the Event Logging mechanism	52
GLOBAL	Define name and sizes of Global shared memory areas	30
HARDWARE	Include Hardware Macro Instructions	26
IIISERVICES	Include MAX III Compatible services	12
INS	Insert IVSYS Macro prototypes	8
INTERRUPT	Define a task scheduling interrupt	43
INTERTASK	Define the Intertask Communication System in this system	16
LOGFORMAT	Specify system events to be immediately formatted by the Event Logging Mechanism	56
LOGPOINT	Specify the system events to be logged by the Event Logging mechanism	54
MACHINE	Define target CPU type	37
MAPS	Define Structure of MAP pool and scheduling algorithm	28
MLSEQU	Select Sequential Task Loader	22
OCPASSWORD	Define Operator Communications Password	23
PARITY	Define CLASSIC 78xy memory parity trap parameters	36
PGM	Program Name	9
PORT	Define an ITC port for a task	17
ROCDIR	Specify a resident OC directive	20
ROCPAC	Specify a package of resident OC directives	21
SERPACK	Specify packages of services	14
SERVER	Define an ITC server name for a task	19
SERVICE	Include a MAX IV service	13
SHLOMO	Reserve space for directory of shared load modules	34

PART 2 - BASIC I/O SYSTEM BLOCK

PART 3 - TASK STRUCTURE BLOCK

PART 4 - FILE MANAGER STRUCTURE BLOCK

TRANSIENT PARTITION DEFINITION

APPENDIX B
STANDARD REX SERVICES AND PACKAGES OF SERVICE

B.1 STANDARD PACKAGE-OF-PACKAGES

Package Name	Included Packages	Characteristics
ALL	NORMAL	Services required for "average" system
	TSCHEDULER	Timer task-scheduling services
	ISCHEDULER	Interrupt task-scheduling services
	XTASK	Extended task services
	GSHARE	Global shared region services
	PSHARE	Private shared region services
	SHMODULE	Shared load module services
NORMAL	MINIMUM	Only the required services
	BTIME	Basic timer services
	DEBUG	Basic debugging services
	XIOS	Extended I/O system services
	TTASK	Transient non-resident task services
	FORTRAN	Service required to support FORTRAN
	BATCH	Services required to support manual batch processing
	ROLL	REX ROLL service
MINIMUM	BIOS	Basic I/O system
	BTASK	Basic resident task services
	OTASK	Basic services to let one task affect another
	SYNTAX	Services for analyzing byte strings and converting resultant parameters to binary numbers
	CONVERT	Services for converting binary numbers into ASCII strings for external representation

B.2 STANDARD PACKAGE-OF-SERVICES

Package	Services
BIOS	READ, BKFILE, BKRECORD, TERMINATE, WRITE, AVFILE, AVRECORD, REWIND, WEOF, HOME, IOWAIT
XIOS	ASSIGN, TASSIGN, TAKE, GIVE
BTASK	MESSAGE, WAIT, EXIT, ABORT, RLINQ, RTILL, STATES
TTASK	ALLOCATE, DEALLOCATE, LOVER, USER, RXLP, INIRES
OTASK	RESUME, ACTIVATE, KILL, TINFO, MODOPT, GETOPT, MODPOP, GETPOP, SETVAR, GETVAR, DELVAR, MARS
*ISCHED or TSCHED or both	CONNECT, THAW, FREEZE, UNCONNECT
XTASK	ESTABLISH, DEESTAB, CHANGE
SYNTAX	GET, COLLECT, ATC, ATN
CONVERT	CTA, BTD, DTD, HEX
BTIME	DELAY, TIME
DEBUG	DEBUG, DUMP, USER, PAUSE, ESTABLISH, DEESTAB, CHANGE
FORTRAN	FORT
BATCH	USER, LOVER, RXLP
GSHARE	IGSHARE, EXTR, ISMODULE
PSHARE	CPSHARE, IPSHARE, EXTR, ISMODULE
SHMODULE	ISMODULE, EXTR

* If either or both of these packages are specified for the system, at least one TIMER and INTERRUPT statement must also be specified in the SYSGEN.

B.3 STANDARD SERVICE-IN-PACKAGES

Service Name	Service Number	Package(s) Name In Which Included	Service Use
READ	Ø	BIOS,MINIMUM,NORMAL,ALL	Read one physical record (block) from an I/O device or file.
WRITE	1	" " " "	Write one physical record (block) to a device or file
REWIND	2	" " " "	Reposition device or file to beginning-of-medium
BKFILE	3	" " " "	Backspace over physical records on device or file until a file mark record or beginning-of-medium is detected.
BKRECORD	4	" " " "	Backspace over physical record (block) on I/O device or file.
AVRECORD	5	" " " "	Advance over a physical record (block) on I/O device or file.
AVFILE	6	" " " "	Advance over physical records on device or file until file mark record is detected.
WEOF	7	" " " "	Write end-of-file physical record on device or file
HOME	8	" " " "	Normalize (initialize) device for subsequent new application usage.
TERMINATE	9	" " " "	Immediately unqueue or terminate device or file operation previously queued.
ASSIGN	#A	XIOS,BIOS,MINIMUM,NORMAL,ALL	Associate a new device or real file with logical file name.

TASSIGN	#B	XIOS,BIOS,MINIMUM,NORMAL,ALL	Test assignment of logical file and return pertinent information about the device or real file assigned to it.
TAKE	#23	" " " " "	Take exclusive use of a device or file or volume on which file is resident.
GIVE	#24	" " " " "	Remove exclusive use of a device or file or volume on which file is resident.
MESSAGE	#1Ø	BTASK,MINIMUM,NORMAL,ALL	Output message to system operator terminal and optionally suspend (HOLD) calling task until operator responds appropriately.
WAIT	#11	" " " "	Suspend (HOLD) calling task until cooperating task or event resumes its execution.
EXIT	#12	" " " "	Terminate execution of calling task and return all resources to system pools gracefully.
ABORT	#13	" " " "	Abnormally terminate execution of calling task and return all returnable resources immediately.
RLINQ	#1D	" " " "	Relinquish execution of calling task till next system event, or declare a system event.
RTILL	#1E	" " " "	Relinquish calling tasks execution until some single valued criteria is met; review criteria on every system event.
STATES	#2Ø	" " " "	Return information to calling task about its various task states (floating point and decimal overflow/ underflow, job error status, et cetera)

ALLOCATE	#29	TTASK,NORMAL, ALL	Allocate pages of memory dynamically to calling task; roll out tasks of lower priority if necessary.
DEALLOCATE	#2A	" " "	Deallocate pages of memory allocated previously to calling tasks, rolling check-pointed tasks if enough memory is now available for them.
RXLP	#2B	" " "	Perform load overlay service infinite steps - using previously obtained or explicitly generated Load Programs.
LOVER	#2D	" " "	Load an overlay into virtual space and memory belonging to calling task.
USERTRAP	#2F	" " "	Cause calling task to trap locally on subsequent violation.
INIRES	#41	" " "	Initialize a named resident directory.
RESUME	#15	OTASK,MINIMUM,NORMAL,ALL	Resume the execution of another suspended task.
ACTIVATE	#16	" " " "	Start the execution of another task.
KILL	#17	" " " "	Abort (abnormally terminate) another active task.
TINFO	#43	" " " "	Return information about the status of the specified task.
CONNECT	#18	ISCHEDULER or TSCHEDULER NORMAL,ALL	Connect independent scheduling mechanism (timer or interrupt) to a task for purposes of subsequently activating it, resuming it, or aborting it.

THAW	#1A	" " " " "	Enable a task scheduler previously connected to a task.
FREEZE	#1B	" " " " "	Disable a task scheduler previously connected to a task.
UNCONNECT	#19		Disable and unconnect an independent scheduling mechanism from a task. Return scheduler to pool.
ESTABLISH	#27	XTASK,DEBUG,ALL	Pre-load and make "resident" an otherwise non-resident task until further notice.
DEESTABLISH	#28	" " "	Make an established task become a normal non-resident task.
CHANGE	#1C		Change the priority of an existing active or scheduled task.
GET	#34	SYNTAX,MINIMUM, NORMAL,ALL	Parse character string and return parameter.
COLLECT	#35	" " "	Parse character string, classify and return numerical or other character.
ATC	#37	" " "	Convert ASCII character string to CAN-code value.
ATN	#38	SYNTAX,MINIMUM, NORMAL,ALL	Convert ASCII character string into binary number. String can be decimal or hexadecimal format.
CTA	#39	CONVERT,MINIMUM, NORMAL,ALL	Convert CAN-code to ASCII character string.
BTD	#3A	" " "	Convert 16-bit binary number to ASCII character string.
DTD	#3C	" " "	Convert 32-bit binary number to ASCII character string, and insert decimal point.

HEX	#3B	" " "	Convert 16-bit binary number to hexadecimal ASCII string.
DELAY	#14	BTIME,NORMAL,ALL	Suspend task or use dedicated timer of task.
TIME	#40	" " "	Get current time value in system clocks.
DEBUG	#1F	DEBUG,NORMAL,ALL	Enter DEBUG processor from task program.
DUMP	#3F	" " "	Dump memory in hexadecimal or decimal printed format.
USER	#2F	" " "	Supply address of local disposition routines for violation of privileged instructions or execution of unimplemented instructions and services.
FORTRAN	#F	FORT,NORMAL,ALL	Set up linkages for a non-resident FORTRAN Run-Time Package (this is a minimal routine provided for compatibility purposes only). See Appendix B-4.
FORR	#F		Set up linkages for a resident FORTRAN Run-Time Package. (If both FORT and FORR are given, FORR will override and the resident Run-Time package will be present). See Appendix B-4.
CPSHARE	#58	PSHARE,ALL	Create private shared region.
IGSHARE	#59	GSHARE,ALL	Insert global shared region.
IPSHARE	#5A	PSHARE,ALL	Insert private shared region.

ISMODULE	#5B	SHMODULE,ALL, GSHARE,PSHARE,ALL	Insert shared load module.
EXTR	#5C	GSHARE,PSHARE, SHMODULE,ALL	Extract global or service shared region.
ROLL	#3E	NORMAL,ALL	Request ROLLER task to roll tasks into or out of main memory.
EVELOG	#3D		Request event to be processed by the Event logging mechanism.

B.4 FORTRAN RESIDENT RUN-TIME PACKAGE

To install the FORTRAN Resident Run-Time Package the user must rearrange the ALB Library. The module FRT:RR must be moved in front of all the other FORTRAN routines used for I/O. The easiest way to do that is to delete modules RUN66 through BIO:A. Below is a job stream that will make FRT:RR the first module on ALB.

```
$JOB
$ASS BI=ALB BO=SCA
$EXE LIB
LNA
REW BI BO
DEL RUN66 BIO:A
COPY
ASS BI SCA BO ALB
COPY
EXIT
$EOJ
```

To install a MAP Ø resident FORTRAN Run-time Package add the following statement in the SYSBLOCK of the user SYSGEN.

```
SERVICE       FORR
```

To unmap the FORTRAN Resident Run-time Package using MP.TSK the user must still rearrange ALB but the user does not include the SERVICE FORR statement in the SYSGEN. Refer to the section on NON-MAP Ø resident code in the MAX IV General Operating System, System Guide Manual.

APPENDIX C
CONTROLLER NAMES, NON-DISC AND DISC DEVICES AND DISC DRIVE TYPES

C.1 STANDARD CONTROLLER NAMES

Controller Name	Controller Description
	* DATA STORAGE DEVICE CONTROLLERS *
CD	Cartridge-style Moving-Head Disc
CX	Cartridge-style Moving Head Disc (double density)
MP	Moving-Head Removable-Pack Disc
MX	Moving-Head Removable-Pack Disc (double density)
FH	Fixed-Head Nonremovable-Pack Disc
FL	Flexible-Media Moving-Head Disc (floppy)
LD	Large-Capacity Moving-Head Disc
LX	CLASSIC Moving-Head Disc
BM	MEMORY+ Bulk-Core-Module Pseudo-Fixed-Head Disc
MT	Magnetic Tape
	* UNIT RECORD DEVICE CONTROLLERS *
TT	Teletype/Terminal/Paper-Tape Reader/Paper-Tape Punch
PP	High-Speed Paper-Tape Punch
CR	Card Reader
DR	DMP Card Reader
CP	Card Punch
LP	Line Printer
DP	DMP Line Printer
LC	Line Printer, Serial (Centronics-style)
VP	Printer/Plotter, Raster-Type (Versatec)
XY	X-Y Plotter
	* PROCESS INPUT/OUTPUT CONTROLLERS *
WS	Wide-Range Solid-State Analog Input Subsystem
IO	IOIS Subsystem (Analog Output)
ID	IOIS Subsystem (Digital Input/Output)
	* 4820 COMPUTER-COMPUTER-LINK CONTROLLER *
CL	4820 Computer-Computer-Link
	* 4523 ASYNCHRONOUS FLOPPY DISC CONTROLLER *
AF	Asynchronous Floppy Disc

C.2 STANDARD NON-DISC DEVICES

TYPE	LOGICAL DEVICE NAME	CONTROLLER NAME	RECORD SIZE IN BYTES	DEFAULT DEVICE OPTION*
Keyboard/printer or keyboard/screen (CRT)	TYn	TTx	82	#24
High or low speed paper-tape reader	PRn	TTx	0(unlimited)	#105
Slow paper-tape punch of ASR terminal device	TPn	TTx	0(unlimited)	#208
Paper-tape punch	PPn	PPx	0(unlimited)	#208
Card reader	CRn	CRx	80	#10
DMP card reader	CRn	DRx	80	#10
Card punch	CPn	CPx	80	#10
Line Printer	LPn	LPx	134	#20
DMP Line Printer	LPn	DPx	134	#20
Centronics Line Printer	LPn	LCx	134	#1820
X-Y Plotters	XYn	XYx	0(unlimited)	#40
Raster plotter/printer (Versatec)	VPn	VPx	0(unlimited)	#60
Magnetic Tape	MTn	MTx	0(unlimited)	#02
CPU-CPU Link	CLn	CLx	0(unlimited)	#00

* Device options are specified by Parameter 4 of the DEVICE statement.

C.3 STANDARD DISC DEVICES

	CONTROLLER NAME	SECTOR SIZE IN BYTES	TRACK SIZE IN SECTORS	CYLINDER SIZE IN TRACKS	TRANSPORT SIZE IN CYLINDERS	DEVICE OPTION
Moving-head cartridge disc	CDx	2ØØ	32	2	2Ø3	#Ø1
Moving-head cartridge disc (double density)	CXx	256	24	2	816	#Ø1
Moving-head removable pack disc	MPx	256	24	2Ø	2Ø3	#Ø1
Moving-head removable pack disc (double density)	MXx	256	24	2Ø	4Ø6	#Ø1
ixed-head disc	FHx	256	32	1	256	#Ø1
Flexible media moving-head disc (floppy)	FLx	256	16	1	77	#Ø1
Large-capacity moving-head disc (333Ø type)	LDx	256	42*	19*	411*	#Ø1
CLASSIC moving-head disc - extended	LXx	256	42*	19*	411*	#Ø1
Bulk Memory	BMx	256	32	1	512	#Ø1
Asynchronous Floppy Disc	AFx	256	15/26	1	77	

These are default values for the CLASSIC and Large Capacity moving-head discs when no type is specified in the DISCCHARACTER SYSGEN statement. (Default to type 1.)

C.4 CLASSIC AND LARGE CAPACITY DISC DRIVE TYPES

TYPE	SECTORS/ TRACK	HEADS	CYLINDERS	TOTAL TRACKS	UNFORMATTED CAPACITY
(0)	All alternate trackable devices				
1	42	19	411	7809	100MB
2	42	19	815	15485	200MB
3	64	19	411	7809	150MB
4	64	19	815	15485	300MB
5	64	19	823	15637	300MB
(8)@	64	4	200	800	16MB
9	64	5	411	2055	40MB
10+	64	5	823	4115	80MB
11*	64	10	823	8230	160MB
12*	64	2	320	640	12MB
13*	64	4	320	1280	24MB
(14)@	64	1	1646	1646	32MB

* Fixed Media.
+ Fixed or Removable Media.
@ Mixed Media - Fixed and Removable Media.
( ) Not available on Large Capacity Disc Controllers.

CLASSIC DISC DRIVE RECOMMENDED PARAMETERS FOR TYPE 0

The Table below uses the following headings:

PHYSICAL CYLINDERS - The total number of cylinders on the physical media.

USABLE CYLINDERS - The PHYSICAL CYLINDERS less the number of cylinders set aside for the spare track pool.

USABLE TRACKS - The number of tracks on the disc as seen by the Operating System. This number is the value to be used in the LOGTRANSPORT statements at SYSGEN. USABLE TRACKS = USABLE CYLINDERS * HEADS.

UNFORMATTED CAPACITY	SECTORS/ TRACK	HEADS	USABLE CYLINDERS	PHYSICAL CYLINDERS	USABLE TRACKS	PHYSICAL TRACKS
.75MB	16	2	80	80	160	160
12MB+	64	2	305	320	610	640
16MB@	64	4	200	206	800	823
24MB*	64	4	305	320	1220	1280
27MB*	32	8	300	310	2400	2480
32MB@	64	2	808	823	1616	1646
48MB	64	5	400	411	2000	2055
80MB+	64	5	805	823	4025	4115
150MB	64	19	405	411	7695	7809
160MB+	64	10	812	823	8120	8230
300MB	64	19	809	815	15371	15485
300MB	64	19	817	823	15523	15637

* Fixed Media.
+ Fixed or Removable Media.
@ Mixed Media - Fixed and Removable Media.

C.5 ASYNCHRONOUS FLOPPY DISC DRIVE TYPES

TYPE	CYLINDERS	HEADS	SECTORS	FORMATTED CAPACITY
1	77	1	15	147,840
2	77	1	26	256,256

APPENDIX D
STANDARD TASK SYSTEM OPTIONS

BATCH (JOB CONTROL) TASK'S SYSTEM OPTIONS

NAME		Bit in OPTION TCB word	PURPOSE - if ON (1)
UØ		Ø	Used by various processors and FORTRAN Run-Time package
U1		1	
U2		2	
U3		3	
U4		4	
U5		5	
U6		6	Use Ø26 card conversion
LO or	U7	7	Permit Listing Output
BO or	U8	8	Permit Binary Output
SC[RATCH] or	U9	9	Use SCRATCH file
HO[LD] or	UA	1Ø	Enter HOLD state at "various" breakpoints
MA[P] or	UB	11	Permit MAP to be output
GO or	UC	12	Continue even if errors exist
AO or	UD	13	Use Alternate File for Commands
HL[OAD] or	UE	14	Place task in HOLD state at completion of each LOAD operation
DU[MP] or	UF	15	Dump body of task to global DO file if task should abort

OC TASK'S SYSTEM OPTIONS

NAME	Bit in OPTION TCB word	PURPOSE - if ON (1)
U0	0	Inhibit two minute time out after the first character has been received
U1	1	Suspend the OC task after the first directive has been completed
U2	2	Inhibit the initial cold start-up of the OC task
U3	3	Never time out OC task
U4	4	Change attentive task name to "X" if "//" typed
U5	5	Do not take exclusive use of the CO file
U6	6	The OC task will NOT print the "OC RETURNED TO DEFAULT" message.

APPENDIX E
DIRECT MEMORY PROCESSOR NUMBERS

MODCOMP - Standard Device Controllers - Relative Dedicated Location**

DEVICE	GROUP	UNIT	NON-VIRTUAL DMP #	CLASS	<-----------Virtual Mode DMP #----------> (Hexadecimal)			
					<-----Standard-----> (12 DMP's)		<-----Special-----> (16 DMP's)	
					PIOP	SIOP	PIOP	SIOP
FH DISC	# Ø	2	2	I	# Ø A	# 1 A	# Ø E	# 1 E
MH DISC	# Ø	1	1	I	# Ø 1	# Ø 1	# Ø D	# 1 D
FLOPPY	# Ø	9	3*	II	# Ø 3*	# 1 3*	not available for Class II	not available for Class II
MAG TAPE (High Per-	# Ø for-	3	3* mance)	II	# Ø 3*	# 1 3*		
MAG TAPE (Low Per-	# Ø for-	4	4 mance)	II	# Ø 4	# 1 4		
PLOTTER/ PRINTER	# Ø	8	7	II	# Ø 7	# 1 7		
.MODAC	various		3*	II	# Ø 3*	# 1 3*		

NOTES:

* If these devices co-exist in a system, only one can use the dedicated location; the others must be assign a different unique DMP number (relative dedicated location). When the PIOP and SIOP are both present, the relative dedicated locations must be unique for all devices --- thereby limiting the total number of DMP channels to 16 instead of the implied 32. Only class 1 device controllers can use the alternate SPECIAL DMP numbers.

** Actual dedicated locations are #6X for Transfer Count (TC) and #7X for Transfer Address (TA) where X is the relative dedicated location. These locations are in actual page zero.

. MODAC - Modular Acquisition Controller

APPENDIX F
EXAMPLE OF A MAX IV SYSGEN

```
TTL            SUPER MAX IV:   GS.IV
INS            MC,IVSYS
SYSID          SUPER,MAX,IV
TASKMASTER     1ØØ,ROUNDROBIN,,25
IIISERVICES
SERPACK        ALL
SERVICE        FORR
INTERTASK      3ØØ,1Ø
PORT           SDF,,SM,,AMA
SERVER         SDF.READ,SDF
PORT           SDI,,SM,,AMA
SERVER         SDF.WRITE,SDI
ROCPAC         STD
INTERRUPT      7
INTERRUPT      8
INTERRUPT      9
INTERRUPT      #A
INTERRUPT      #B
TIMERS         4
DCTIMERS       4
SIMULATION     3X
HARDWARE       3X
GLOBAL         SMGLOB,1
GLOBAL         LRGLOB,2
SHLOMO         3
SHPRIVATE      3,12
CLOCK          2,CALENDAR,DISPLAY,UTILIZATION

DIRECTORY      BM,3Ø
SYSBUG         64,BOTH,MT4
ENDSYSTEM
TTL            SUPER MAX IV:   GI.IV
INS            MC,IVIOS
CONTROLLER     CD,1,1,TIMED
CONTROLLER     FH,2,#A,TIMED
CONTROLLER     LX,#3,3,TIMED
CONTROLLER     MT,4,4,TIMED
CONTROLLER     CR,5
CONTROLLER     CX,6,#Ø,TIMED
CONTROLLER     LP,7
CONTROLLER     VP,8,7,,,,67
CONTROLLER     PP,9,,TIMED
CONTROLLER     TT,#A,,TIMED
CONTROLLER     LC,#B
CONTROLLER     FL,#C,5,TIMED
SYMCONTROL     S
SYMCONTROL     RDS
FMTRANSPORT    PMØ,Ø,CD
LOGTRANSPORT   MØ,PMØ,Ø,4Ø6,NOF
DISCDEVICE     TPØ,MØ,2,4,2ØØ,4Ø4
FMTRANSPORT    PM1,1,CD
LOGTRANSPORT   M1,PM1,Ø,4Ø6,FMP,,1ØØ
FMTRANSPORT    PFØ,Ø,FH
```

```
LOGTRANSPORT FØ,PFØ,Ø,1Ø5Ø,FMP
FMTRANSPORT  PTØ,Ø,CX
LOGTRANSPORT TØ,PTØ,Ø,816,FMP
LOGTRANSPORT T1,PTØ,816,816,FMP
DEVICE       MT1,MT
DEVICE       MT2,MT
DEVICE       MT3,MT
DEVICE       MT4,MT
DEVICE       CRP,CR
DEVICE       LPP,LP
DEVICE       VP,VP,134
DEVICE       PP,PP
DEVICE       TY,TT
DEVICE       PR,TT
DEVICE       TP,TT
DEVICE       LC,LC,134
FMTRANSPORT  PPØ,Ø,FL
LOGTRANSPORT PØ,PPØ,Ø,77,FMP
FMTRANSPORT  PP1,1,FL
LOGTRANSPORT P1,PP1,Ø,77,FMP
FMTRANSPORT  PLØ,Ø,LX,,,,,,,3
LOGTRANSPORT LØ,PLØ,Ø,78Ø9,FMP
FMTRANSPORT  PL1,1,LX,,,,,,,3
LOGTRANSPORT L1,PL1,Ø,78Ø9,FMP
DEVICE       LP,S,134,#ØØ2Ø
DEVICE       TY3,S,74,#332Ø
DEVICE       CR,RDS,8Ø,#ØØ1Ø
TLPDEFINE    TPD,3ØØ
ENDIOS       MESSAGES,STALLS
TTL          SUPER MAX IV:  GT.IV
INS          MC,IVTASK
ATTENTASK    B
TRANSIENT    2Ø
IONODE       1Ø,1ØØ
PRESCHEDULE  RDS,,SM
PRESCHEDULE  S,,SM
PRESCHEDULE  B,,BM
TASK         X,Ø,Ø,SM
PECULIAR     UNABORTABLE
LOGFILE      CO,TY
LOGFILE      OC,TY
LOGFILE      DO,LP
LOGFILE      LM,QLM
LOGFILE      SM,QSM
LOGFILE      USM,QSM
LOGFILE      LMA,DNQ
LOGFILE      ALB,DNE
LOGFILE      RO,QRO
LOGFILE      LB,NO
LOGFILE      AFM,ANU
LOGFILE      BM,QBM
LOGFILE      JC:,CR
LOGFILE      FM,NO
LOGFILE      RAD,NO
LOGFILE      LMT,QTM
LOGFILE      TPD,TPØ
LOGFILE      CDF,ACF
LOGFILE      SAF,AAF
LOGFILE      AAF,AAA
LOGFILE      FMM,MMC
LOGFILE      JMF,QSM
LOGFILE      JM,QSM
LOGFILES     5
FINTASK      X
TASK         OC,1,Ø,SM
PECULIAR     UNABORTABLE,AUTOSTART
STACKS       16Ø,16
LOGFILE      CO,TY
LOGFILE      OC,TY
FCBDEFINE    1
SYSPAGES     3
FINTASK      OC
FMGRX
ROLLER
ENDTASKS
TTL          SUPER MAX IV:  GF.IV
INS*         FMM,FMSMAC
RESSERVICE   ALL
MAXNAME      39,13
TCLASS       CD,M1
TCLASS       FL,PØ,P1
TCLASS       LX,LØ,L1
FDLDEFAULTS
FSG          1-TRACK
ENDFDL
FMCATALOG    SCRVOL,CDF,1ØØ
FMSECURITY   MSTRID
FMAUDITOR    SAF,AAF,AF,RA,NS
AUDITPOINT   ALL,AS
FMEND
```

APPENDIX G
EXAMPLE OF TRANSIENT LDT DEFINITIONS

```
$JOB PUT TRANSIENT DEFINITIONS ON TPØ
$ASS BO TPØ
$EXE M5A,,NOLO
       INS            MC,IVIOS
       TPDDEVICE      LX              *DEFINE DEFINITION FILE DEVICE
*
*      TRANSIENT LDT DEFINITIONS FOR TRANSPORT LØ
*
       DISCCHARACTER LO,LX,,,,,NOC,NOE,MOV,8 ** LOGICAL UNIT Ø **
       LET            CTNLØ,2
       DISCDEVICE     ALS,LØ,34,2Ø
       LET            CTNLØ,38
       DISCDEVICE     QBM,LØ,65,1,Ø
       DISCDEVICE     QSM,LØ,2Ø,1,Ø
       DISCDEVICE     QLM,LØ,7,1,Ø
       DISCDEVICE     ALB,LØ,23,2Ø
       DISCDEVICE     AJC,LØ,8,2Ø
       DISCDEVICE     AMC,LØ,34,2Ø
       DISCDEVICE     AL4,LØ,23,2Ø
       DISCDEVICE     QLT,LØ,17,1,Ø
       DISCDEVICE     DLD,LØ,2,2Ø
       DISCDEVICE     DST,LØ,13,2Ø
       DISCDEVICE     AMX,LØ,9,2Ø
       DISCDEVICE     BMS,LØ,24,2Ø,,235
       DISCDEVICE     BSL,LØ,15,2Ø
       DISCDEVICE     BSW,LØ,22,2Ø
       DISCDEVICE     BJM,LØ,2,1,Ø
       DISCDEVICE     BSC,LØ,28,2Ø
       DISCDEVICE     BSA,LØ,14,2Ø
       DISCDEVICE     BSB,LØ,29,2Ø
       DISCDEVICE     BS,LØ,71,2Ø,,298
       DISCDEVICE     BJW,LØ,5,2Ø
       DISCDEVICE     SPØ,LØ,12,2Ø
       DISCDEVICE     ADA,LO,4,2Ø
       DISCDEVICE     ADB,LØ,1Ø,2Ø
       DISCDEVICE     BMC,LØ,1Ø,2Ø,,39Ø
       DISCDEVICE     AD,LØ,14,2Ø,,386
       DISCDEVICE     ADC,LØ,1ØØ,2Ø
       DISCDEVICE     ADD,LØ,1ØØ,2Ø
       DISCDEVICE     ADE,LØ,1ØØ,2Ø
       DISCDEVICE     ADF,LØ,1ØØ,2Ø
*
*      TRANSIENT LDT DEFINITIONS FOR TRANSPORT L1
*
       DISCCHARACTER L1,LX,,,,,NOC,NOE,MOV,8 ** LOGICAL UNIT 1 **
       DISCDEVICE     AD1,L1,1ØØ,2Ø
       DISCDEVICE     AD2,L1,1ØØ,2Ø
       DISCDEVICE     AD3,L1,1ØØ,2Ø
       DISCDEVICE     AD4,L1,1ØØ,2Ø
       DISCDEVICE     AD5,L1,1ØØ,2Ø
       DISCDEVICE     AD6,L1,1ØØ,2Ø
       DISCDEVICE     AD7,L1,1ØØ,2Ø
       DISCDEVICE     AD8,L1,98,2Ø
```

```
*
*       TRANSIENT LDT DEFINITIONS FOR TRANSPORT FØ
*
        DISCCHAR      FØ,FH
        DISCDEVICE    FØ4,FØ,175,4,256,Ø
        DISCDEVICE    FØ1,FØ,175,1,Ø,Ø
        DISCDEVICE    FHA,FØ,21,,,Ø
        DISCDEVICE    FHB,FØ,21,,,21
        DISCDEVICE    FHI,FØ,42,,,Ø
        DISCDEVICE    FHC,FØ,21,,,42
        DISCDEVICE    FHD,FØ,21,,,63
        DISCDEVICE    FHJ,FØ,42,,,42
        DISCDEVICE    FHM,FØ,84,,,Ø
        DISCDEVICE    FHE,FØ,21,,,84
        DISCDEVICE    FHF,FØ,21,,,1Ø5
        DISCDEVICE    FHK,FØ,42,,,84
        DISCDEVICE    FHG,FØ,21,,,126
        DISCDEVICE    FHH,FØ,28,,,147
        DISCDEVICE    FHL,FØ,49,,,126
        DISCDEVICE    FHN,FØ,91,,,84
        DISCDEVICE    FHO,FØ,175,,,Ø
*
*       TRANSIENT LDT DEFINITIONS FOR TRANSPORT BØ
*
        DISCCHAR      BØ,BM
        DISCDEVICE    BAØ,BØ,25
        DISCDEVICE    BA1,BØ,25
        DISCDEVICE    BA2,BØ,25
        DISCDEVICE    BA3,BØ,25
        DISCDEVICE    BAA,BØ,5Ø,,,Ø
        DISCDEVICE    BAB,BØ,5Ø
        DISCDEVICE    BAC,BØ,1ØØ,,,Ø
*
*       TRANSIENT LDT DEFINITIONS FOR TRANSPORT PØ
*
        DISCCHAR      PØ,FLØ
        DISCDEVICE    PØ4,PØ,77,4,256,Ø
        DISCDEVICE    PØ1,PØ,77,1,Ø,Ø
        DISCDEVICE    PAA,PØ,9,,,Ø
        DISCDEVICE    PAB,PØ,9,,,9
        DISCDEVICE    PAI,PØ,18,,,Ø
        DISCDEVICE    PAC,PØ,9,,,18
        DISCDEVICE    PAD,PØ,9,,,27
        DISCDEVICE    PAJ,PØ,18,,,18
        DISCDEVICE    PAM,PØ,36,,,Ø
        DISCDEVICE    PAE,PØ,9,,,36
        DISCDEVICE    PAF,PØ,9,,,45
        DISCDEVICE    PAK,PØ,18,,,36
        DISCDEVICE    PAG,PØ,9,,,54
        DISCDEVICE    PAH,PØ,14,,,63
        DISCDEVICE    PAL,PØ,23,,,54
        DISCDEVICE    PAN,PØ,41,,,36
        DISCDEVICE    PAO,PØ,77,,,Ø
```

```
*
*       TRANSIENT LDT DEFINITIONS FOR TRANSPORT P1
*
        DISCCHAR      P1,FLØ
        DISCDEVICE    P14,P1,77,4,256,Ø
        DISCDEVICE    P11,P1,77,1,Ø,Ø
        DISCDEVICE    PBA,P1,9,,,Ø
        DISCDEVICE    PBB,P1,9,,,9
        DISCDEVICE    PBI,P1,18,,,Ø
        DISCDEVICE    PBC,P1,9,,,18
        DISCDEVICE    PBD,P1,9,,,27
        DISCDEVICE    PBJ,P1,18,,,18
        DISCDEVICE    PBM,P1,36,,,Ø
        DISCDEVICE    PBE,P1,9,,,36
        DISCDEVICE    PBF,P1,9,,,45
        DISCDEVICE    PBK,P1,18,,,36
        DISCDEVICE    PBG,P1,9,,,54
        DISCDEVICE    PBH,P1,14,,,63
        DISCDEVICE    PBL,P1,23,,,54
        DISCDEVICE    PBN,P1,41,,,36

*
*                     411 CYLINDER 4138/333Ø DRIVE Ø
*
        DISCCHAR      LØ,LD
        DISCDEVICE    LØØ,LØ,6ØØØ,14,,Ø
        DISCDEVICE    LØ1,LØ,3ØØØ,14,,Ø
        DISCDEVICE    LØ2,LØ,3ØØØ,14,,3ØØØ
        DISCDEVICE    LØ3,LØ,15ØØ,14,,Ø
        DISCDEVICE    LØ4,LØ,15ØØ,14,,15ØØ
        DISCDEVICE    LØ5,LØ,15ØØ,14,,3ØØØ
        DISCDEVICE    LØ6,LØ,15ØØ,14,,45ØØ
        DISCDEVICE    LØ7,LØ,75Ø,14,,Ø
        DISCDEVICE    LØ8,LØ,75Ø,14,,75Ø
        DISCDEVICE    LØ9,LØ,75Ø,14,,15ØØ
        DISCDEVICE    LØA,LØ,75Ø,14,,225Ø
        DISCDEVICE    LØB,LØ,75Ø,14,,3ØØØ
        DISCDEVICE    LØC,LØ,75Ø,14,,375Ø
        DISCDEVICE    LØD,LØ,75Ø,14,,45ØØ
        DISCDEVICE    LØE,LØ,75Ø,14,,525Ø
*
*                     411 CYLINDER 4138/333Ø DRIVE 1
*
        DISCCHAR      L1,LD
        DISCDEVICE    L1Ø,L1,6ØØØ,14,,Ø
        DISCDEVICE    L11,L1,3ØØØ,14,,Ø
        DISCDEVICE    L12,L1,3ØØØ,14,,3ØØØ
        DISCDEVICE    L13,L1,15ØØ,14,,Ø
        DISCDEVICE    L14,L1,15ØØ,14,,15ØØ
        DISCDEVICE    L15,L1,15ØØ,14,,3ØØØ
        DISCDEVICE    L16,L1,15ØØ,14,,45ØØ
        DISCDEVICE    L17,L1,75Ø,14,,Ø
        DISCDEVICE    L18,L1,75Ø,14,,75Ø
        DISCDEVICE    L19,L1,75Ø,14,,15ØØ
        DISCDEVICE    L1A,L1,75Ø,14,,225Ø
        DISCDEVICE    L1B,L1,75Ø,14,,3ØØØ
        DISCDEVICE    L1C,L1,75Ø,14,,375Ø
        DISCDEVICE   L1D,L1,750,14,,4500
        DISCDEVICE   L1E,L1,750,14,,5250
        TPDEND                         *MARK END OF DEFINITIONS
$$
$WEOF BO
$EOJ
```

APPENDIX H
EXAMPLES OF POWER FAIL RESTART

The MAX IV Operating System does not perform a warm restart. Routines are supplied to help users restart MAX IV after a power failure. These routines, along with the source for ASAL, are cataloged on USL (partition BSL) under their program names.

The power fail restart routines cause MAX IV to reload using a special version of the Auto-Reboot Stand-Alone Linking Loader (ASAL). ASAL sets some power fail bits and the system is started. User tasks can then test and reset a bit indicating that a power failure has occurred (Bit 14 of LCBSTA).

The following modification to ASAL Revision K.1 is required so that ASAL automatically reloads MAX IV.

If "HØ4,,2Ø" is not to be used to boot the MAX IV System, it is necessary to change the statement at label CBUF in ASAL as follows:

```
*THE FOLLOWING CONSTANT SHOULD BE IDENTICAL TO WHAT YOU WOULD
*TYPE ON THE CONSOLE TO LOAD YOUR MAX IV SYSTEM
CBUF DFC       "HØ4,,2Ø,3",Ø  LOAD FROM 4138 TYPE CONTROLLERS
         --OR--
CBUF DFC       "HØ4,,2Ø",Ø    4176/4176A/418Ø/4185 TYPE CONTROLLER
         --OR--
CBUF DFC       "MØ4",Ø        LOAD FROM DIABLO/2ØØTPI
         --OR--
CBUF DFC       "FØ4",Ø        FIXED HEAD/BULK MEMORY
```

The modified ASAL should be assembled with M5A. Record the track address that the assembled output is written to for use in STEP 5 of the ASAL Dump Procedure below. Write an End-of-File.

Two tracks must be dedicated on the users system disc for this modified ASAL. Tracks 2 and 3 are suggested but not required.

WARNING

> Tracks dedicated for modified ASAL should not be defined as available to the Operating System.Accidential erasure could go undetected until a power failure occurs.

NOTE: Track number is equal to cylinder number multiplied by heads per cylinder plus head number. Does not apply to Lark Disc.

To place the bootable ASAL on the disc in the proper format at the appropriate place, perform the following steps.

1. Press Master Clear
2. Press Fill
3. Press Run
   (SAL startup message appears)
4. Type BIAS #E7ØØ

5. Load Modified ASAL from track that it was assembled on

   (Example: H0,325,20)
   SATISFIED message will appear.

6. Press Halt
7. Press Master Clear
8. Place Device Address of System Disc in Console Switches
9. Press Fill
10. Set Register 6 to #F400 (High dump address)
11. Set Register 7 to #E700 (Low dump address)
12. Set Register 5 Bits 0-7 to the sector (normally 0) and
    Bits 8-15 to the Head number that is to
    accept the dump.
13. Set the Cylinder Number in Bits 0-7 of the Console Switches of destination Cylinder for the modified ASAL.
14. Set Console Switches 8 (Halt after Dump), 9 (Dump Function) and 10 (Sector and Head in R5)
15. Press Run

If an error is detected the Dump program retries automatically until an error free Dump is completed. The CPU will then Halt.

These special handlers are cataloged on the USL file under the following names:

SPDIAB	DIABLO/200TPI Handler
SP4138	4138 Type Controller Handler
SP4176	4176/4176A/4180/4185 Type Controller Handler

The Equates for the Cylinder, Head and Sector should be modified for compatability with the modified ASAL dump location. The Device Address and DMP Equates can also be changed if necessary to those of the System Disc. Incorporate this handler into the system by placing it behind the remainder of the SYSGEN Source. Perform a normal SYSGEN with this handler included and the procedure is complete.

This reboot cannot be successful unless the following conditions were satisfied:

1. Memory remained stable during the outage. (Battery backup functioned properly.)

2. The Disc Controller returned to a ready and operable state when the power returned.

APPENDIX I
RESIDENT OC DIRECTIVES

I.1 RESIDENT OC MODULES

Module Name	Includes Directives	Module Size
ABO	A,ABO,H,HOL,K,KIL,LET,R,REL,RES,SAM,VAR	#122
ADU	ADU	#1B4
ACT	ACT,CHA,DEE,E,EST,EXE	#7D
APT	APT	#CD
ARC	ARC	#CB
ASS	ASS,DEF	#58
AUD	AUD	#D3
AVA	AVA	#A5
A$I	A$I	#14C
BIA	BIA,MAC,MIN,MOD,MOP	#14D
BRO	BRO	#119
CID	CID	#117
CNV	CNV	#DØ
COM	COM	#2ØØ
C$M	C$M	#1D5
CON	CON,REC,RET	#1BC
CPD	CPD	----
DCI	DCI	#48
DEL	DEL	#47
DEV	DEV	#44
DIS	DAC,DIN,DIS,DOP,TAC,TIN,TOP,TYP	#13A
D$G	D$G	#A5
D$P	D$P	#9Ø
D$1	D$1	#18D

D$3	D$3	#AC	
D$4	D$4	#D5	
D$5	D$5	#BØ	
ECM	ECM	#19D	|
EVE	EVE	#122	
E$M	E$M	#B3	
E$O	E$O	#3B	|
FCB	FCB	#AØ	|
FOL	FOL	#BF	|
GSR	GSR	#1ØF	|
HIS	HIS	#79	|
INF	I,INF	#EB	|
INT	INT	#2D	
ISX	ISX	#58	
I$N	I$N	#8A	|
KIN	KIN	#19	|
LDT	LDT	#73	|
LEN	LEN,WID	#91	|
LOG	LOG	#192	|
LTT	LTT	#76	|
L$G	L$G	#16D	|
MES	MES	#189	|
MPI	MPI	#1C4	|
MPL	MPL	#A3	|
MSG	MSG	#128	|
NDP	NDP	#E2	|
NUM	NUM,N	#F1	
OCR	OCR	#A5	
ON	OFF,ON	#1D6	|

OPT	OPT,POP	#A7
PAS	PAS	#17
PAU	PAU	#EF
PIP	PIP	#32
PRT	PRT	#1Ø1
PWR	PWR	#8B
QUE	QUE	#35
RBL	RBL	#AØ
REW	AVF,AVR,BKF,BKR,HOM,REW,TER,WEO	#8B
ROC	ROC	#1C
ROL	ROL	#AC
SEA	SEA	#92
SHA	SHA,WAT	#DA
SPL	SPL	#B9
SRG	SRG	#122
SVR	SVR	#A6
TAK	TAK,GIV	#86
TBP	TBP	#96
TCL	TCL	#C3
TCT	TCT	#63
TIM	DAT,TIM	#B2
TME	TME,MEM	#F1
TMR	TMR	#46
TOL	TOL	#41
TRA	TRA	#1F5
T$B	TCB,PRO,REG,MAP,STK,VEX,LOX,EXT, SHR,BUF,FAT,NOD	#3A
T$A	T$A	#FB
T$I	T$I	#EA

T$Ø	T$Ø	#F4
T$1	T$1	#DD
T$2	T$2	#13D
UNC	UNC,FRE,THA	#35
VIE	VIE,VIN,VOP	#34
WAC	WAC	#11B
WHO	WHO	#1Ø5
WTH	WTH	#1CC

I.2 RESIDENT OC PACKAGES

Package Name	Includes Modules	Package Size
BIO[S]	ASS,ON,REW,TAS,DCI,A$I, TAK,LEN	#5BD
DEB[UG]	PAU,VIE,ADU,HIS,SEA,SHA	#4BC
ITC	MES,PRT,SVR,CID	#447
TAS[K]	ABO,ACT,OPT	#246
TIM[E]	TIM	#B2
TYP[E]	BIA,DIS,E$M,INF,CNV,NUM,OCR,TCL, WTH,TOL	#95B
FMG	APT,ARC,AUD,DP,EO,FCB,FOL,WAC	#5BØ
MAG[IC]	AVA,BRO,MSG,WHO	#3EB
SYS[TEM]	TME,MPL,NDP,PAS,QUE,RBL,ROC, ROL,SPL,SRG,TBP	#69B
CON[NECT]	CON,DEL,UNC,TMR,TI,INT,IN	#41F
DEV[ICE]	DEV,D$1,D$3,D$4,D$5,DG, ISX,LDT,TCT,LTT,COM,CM	#A2Ø
CLA[SSIC]	ECM,GSR,PIP,PWR	#339
LOG[GING]	LOG,L$G,EVE	#3B2
TCB	TB,TØ,T$1,T$2,KIN	#361

I.3 RESIDENT PACKAGE OF PACKAGES

Package Name	Includes Packages	Package Size
ALL	BIO,DEB,ITC,TAS,TIM,TYP, FMG,SYS,CON,DEV,CLA,LOG, TCB,MAG	#43D4
STD	TAS,BIO,TYP,ITC,DEB,TIM	#1B13
SAT[ELLITE]	TAS,TYP,TCB,ITC,TIM,BIO, ISX,LDT	#1A83

APPENDIX J
MAX IV SYSTEM MEMORY REQUIREMENTS

This appendix contains information regarding the approximate sizes of a basic MAX IV Rev. I.0 system plus additional memory usage when certain statements and options are added to the SYSGEN source. Note: all numbers are stated in decimal unless preceded by a pound sign (#), which indicates hexadecimal.

J.1 BASE MAX IV SYSGEN -- SIZE: 18,420 WORDS

```
TTL          SYSTEM BLOCK
PGM          SYSTST
INS          MC,IVSYS
CLOCK        2
ENDSYSTEM
TTL          BIOS BLOCK
INS          MC,IVIOS
ENDIOS
TTL          TASK BLOCK
INS          MC,IVTASK
TRANSIENT    1
IONODES      1,1
TASK         X
LOGFILE      CO,TY,TY
LOGFILE      OC,TY,TY
FINTASK      X
TASK         OC
LOGFILE      CO,TY,TY
LOGFILE      OC,TY,TY
FINTASK      OC
ENDTASKS
```

J.2 BASE MAX IV FILE MANAGER SYSGEN (MAP 0 RESIDENT) -- SIZE: 29,981 WORDS

```
     TTL           SYSTEM BLOCK
     PGM           SYSTST
     INS           MC,IVSYS
     CLOCK         2
     ENDSYSTEM
*
     TTL           BIOS BLOCK
     INS           MC,IVIOS
     ENDIOS
*
     TTL           TASK BLOCK
     INS           MC,IVTASK
     TRANSIENT     1
     IONODES       1,1
     FMGRX
*
     TASK          X
     LOGFILE       CO,TY,TY
     LOGFILE       OC,TY,TY
     FINTASK       X
     TASK          OC
     LOGFILE       CO,TY,TY
     LOGFILE       OC,TY,TY
     FINTASK       OC
     ENDTASKS
*
     TTL FILE MANAGER SYSTEM CONFIGURATION
     INS           MC,FMSMAC
     RESSERVICE    ALL
     FMEND
```

J.3 BASE MAX IV FILE MANAGER SYSGEN (MAPPED) -- SIZE: 20,491 WORDS

```
         TTL          SYSTEM BLOCK
         PGM          SYSTST
         INS          MC,IVSYS
         CLOCK        2
         ENDSYSTEM
*
         TTL          BIOS BLOCK
         INS          MC,IVIOS
         ENDIOS
*
         TTL          TASK BLOCK
         INS          MC,IVTASK
         TRANSIENT    1
         IONODES      1,1
         FMGRX
*
         TASK         X
         LOGFILE      CO,TY,TY
         LOGFILE      OC,TY,TY
         FINTASK      X
         TASK         OC
         LOGFILE      CO,TY,TY
         LOGFILE      OC,TY,TY
         FINTASK      OC
         ENDTASKS
*
         TTL FILE MANAGER SYSTEM CONFIGURATION
         INS          MC,FMSMAC
         RESSERVICE   MAPPED
         FMEND
```

J.4 SYSGEN STATEMENTS AND MEMORY REQUIREMENTS

STATEMENT OR OPTION	MEMORY IN WORDS ADDED/DELETED
BASIC MAX IV SYSTEM:	18,420
BASIC MAX IV SYSTEM WITH A FUNDAMENTAL FILE MANAGER BLOCK RESIDENT OR MAPPED:	 29,981 20,491
FOR ROUNDROBIN TASKMASTER	+ 76
FOR IIISERVICES	+ 13
FOR SERPACK STATEMENTS ADD:	
ALL	+ 6131
MINIMUM	+ 2058
NORMAL	+ 4012
SIMULATION 3X OR FLOATING POINT SOFTWARE SIMULATION	 + 606
HARDWARE 3X FOR HARDWARE FLOATING POINT OVERFLOW/UNDERFLOW ROUTINES	 + 121
GLOBAL FOR GLOBAL ENTER 8 X NUMBER OF GLOBAL STATEMENTS	 +______
FOR SHRPRIVATE OR SCHLOMO OR DIRECTORIES	+ 157
ENTER 8 X NUMBER OF SHPRIVATE REGIONS (PARM. 1)	+______
ENTER 9 X NUMBER OF SHLOMO REGIONS (PARM. 1)	+______
ENTER 7 + 4 X DIRECTORY (PARM. 2) FOR EACH DIRECTORY STATEMENT	+______
TOCLOCK ENTER 8 X NUMBER OF ENTRIES SPECIFIED	 +______
TIMER ENTER 11 X NUMBER OF TIMER STATEMENTS	 +______
*INTERRUPT ENTER 13 X NUMBER OF INTERRUPT STATEMENTS	 +______
DCINTERRUPTS ENTER 11 X NUMBER OF DCINTERRUPTS + SIZE OF USER CODED MODULES	 +______
*DCTIMERS ENTER 4 X NUMBER OF DCTIMERS + SIZE OF USER CODED MODULES	 +______
SYSBUG ENTER THE FOLLOWING FOR THE SYSBUG STATEMENTS:	
BASIC SIZE	+ 145
VDUMP	+ 965
ADUMP	+ 1530
STACK SIZE (PARM. 1)	+______

* REQUIRES SERPACK ALL

```
EVENTLOGGING
FOR EVENTLOGGING ENTER:
              BASIC SIZE                                           +   867
              2Ø X NUMBER OF NODES (PARM. 8)                       +______
              FORMATTING TO CONSOLE (PARM. 12, OPTIONAL)           +   675

INTERTASK
FOR INTERTASK ADD:
              RESIDENT                                             +  3698
              MAPPED                                               +   71Ø
                           9 X NUMBER OF PORT STATEMENTS           +_______
                           8 X NUMBER OF SERVER STATEMENTS         +_______

CALARM
FOR CALARM ADD 41 X NUMBER OF CALARM STATEMENTS AND SIZES
OF USER CODED ROUTINES (PARM. 4)

FOR RESIDENT OC, ADD 29Ø + SIZE OF EACH
OC MADE RESIDENT

SELECT APPROPRIATE I/O HANDLER:
CONSOLE TY HANDLER                                                 +   37Ø
CARD READER HANDLER                                                +   293
LINE PRINTER HANDLER                                               +   332
DISC COMMON ROUTINE                                                +   134
MH DISC 2ØØ TPI                                                    +   339
4138 DISC HANDLER                                                  +   8Ø1
4176 DISC HANDLER                                                  +   852
ASYNC FLOPPY HANDLER                                               +  1Ø55
ASYNC COMMUNICATIONS HANDLER (W/O CP)                              +  1Ø49
                             (WITH CP)                             +  1Ø87
MAG TAPE HANDLER                                                   +   352
BULK MEMORY HANDLER                                                +   387
CP COMMON ROUTINES                                                 +   4ØØ
19Ø7 COMMON ROUTINES                                               +    8Ø
BISYNC COMMUNICATIONS HANDLER (W/O CP)                             +   763
                              (WITH CP)                            +   632
SYNC COMMUNICATIONS HANDLER                                        +   64Ø
4828 STANDARD LINK HANDLER                                         +   4Ø1
1939 SDLC/HDLC HANDLER                                             +   734
482Ø COMPUTER LINK HANDLER                                         +   137
WIDE RANGE SOLID STATE HANDLER                                     +   13Ø
IOIS HANDLER                                                       +   125

FOR NON-COMMUNICATIONS:
              ENTER NUMBER OF CONTROLLERS X 85                     +______
              ENTER NUMBER OF LOGTRANS X 5Ø                        +______
              ENTER NUMBER OF DEVICES + DISC DEVICES X 12          +______

FOR COMMUNICATIONS:
              ENTER NUMBER OF HALF DUPLEX LINES X 6Ø               +______
              ENTER NUMBER OF FULL DUPLEX LINES X 132              +______
```

```
-------------------------------------------------------------------------
|FOR SYMBIONTS:                                                          |
|              ENTER NUMBER OF SYMCONTROLLERS X 17              +______  |
|                                                                        |
|              ENTER TLPDEFINE PARM. 2 X 13                     +______  |
|                                                                        |
|              ENTER LINEMONITOR X 16                           +______  |
|                                                                        |
|              ENTER PRESCHEDULES X 5                           +______  |
|                                                                        |
|              ENTER TRANSIENTS X 6Ø                            +______  |
|                                                                        |
|              ENTER IONODES (PARM. 1 + PARM 2) X 26 + 15       +______  |
|                                                                        |
|ROLLER                                                                  |
|FOR ROLLER ADD:                                                         |
|              RESIDENT                                         +  2287  |
|              OR MAPPED                                        +   39Ø  |
|                                                               -------  |
|                                                                        |
-------------------------------------------------------------------------
```

INDEX

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 3624 FT. LAUDERDALE, FL 33309

POSTAGE WILL BE PAID BY ADDRESSEE

MODULAR COMPUTER SYSTEMS
1650 W. McNAB ROAD
P.O. BOX 6099
FT. LAUDERDALE, FLORIDA 33310

At..ntion: TECHNICAL PUBLICATIONS, M.S. #85

Fold

MODCOMP®

Please comment on the publication's completeness, accuracy, and readability. We also appreciate any general suggestions you may have to improve this publication.

If you found errors in this publication, please specify the page number or include a copy of the page with your remarks.

Your comments will be promptly investigated and appropriate action will be taken. If you require a written answer, please check the box and include your address below.

□

Comments: ______________________________

Manual Title ______________________________

Manual Order Number ______________________ Issue Date ______________

Name ______________________ Position ______________

Company ______________________________

Address ______________________________

______________________________ Telephone (   ) ______________

MCS 148D (1/83)

MODCOMP

Corporate Headquarters:
MODULAR COMPUTER SYSTEMS, Inc., 1650 West McNab Road, P.O. Box 6099, Ft. Lauderdale, FL 33310, Tel: (305) 974-1380, TWX: 310-372-7837

International Headquarters:
MODULAR COMPUTER SERVICES, Inc., The Business Centre, Molly Millars Lane, Wokingham, Berkshire, RG11 2JQ, UK, Tel: 0734-786808, TLX: 851849149

Latin American Sales Headquarters:
MODULAR COMPUTER SYSTEMS, Inc., 1650 West McNab Road, P.O. Box 6099, Ft. Lauderdale, FL 33310, Tel: (305) 975-6562, TLX: 3727852

Canadian Headquarters:
MODCOMP Canada, Ltd., 400 Matheson Blvd. East, Unit 24, Mississauga, Ontario, Canada L4Z 1N8, Tel: (416) 890-0666, TELEX: 06-961279

SALES & SERVICE LOCATIONS THROUGHOUT THE WORLD

"The technical contents of this document, while accurate as of the date of publication, are subject to change without notice."

Printed in the U.S.A.